

**10/562592**

**IAP20 Rec'd PCT/PTO 28 DEC 2005**

PATENT APPLICATION  
ATTORNEY DOCKET NO. 04995.281001

**APPLICATION  
FOR  
UNITED STATES LETTERS PATENT**

**TITLE: INFORMATION PROCESSING DEVICE,  
INFORMATION PROCESSING SYSTEM,  
OPERATING ARTICLE, INFORMATION  
PROCESSING METHOD, INFORMATION  
PROCESSING PROGRAM, AND GAME SYSTEM**

**APPLICANTS: Hiromu UESHIMA, Keiichi YASUMURA  
and Mitsuru OKAYAMA**

**22511**

PATENT TRADEMARK OFFICE

"EXPRESS MAIL" Mailing Label Number: EV710215901US  
Date of Deposit: DECEMBER 28, 2005

情報処理装置、情報処理システム、操作物、情報処理方法、情報処理プログラム、及び、ゲームシステム

5

## 技術分野

本発明は、オペレータに保持されて動きが与えられる操作物を、ストロボスコープを用いて検出した結果に基づいて、ディスプレイ装置に画像を表示する情報処理装置及びその関連技術に関する。

10

## 背景技術

日本国特開2003-79943号公報(図1、図3)に記載された従来の画像生成システムについて、図面を用いて説明する。

図65は、従来の画像生成システムの説明図である。図65に示すように、検知面形成枠1000内に、二次元の検知面1100が形成される。検知面形成枠1000の辺sd1の2つ角部には、センサs1、s2が設けられる。

15

センサs1は、発光部及び受光部を有している。発光部は、角度 $\theta_1$ が0度～90度の間で赤外線を出し、受光部は、その戻り光を検知する。被検出物である操作物に反射部材が取り付けられているため、その反射部材により反射された赤外線が、受光部により受光される。センサs2についても同様である。

センサs1によって受光された結果は、結像im1として得られる。センサs2によって受光された結果は、結像im2として得られる。結像im1、im2では、被検出物としての操作物が、検知面1100を横切ると、操作物に遮光されない部分が影となって現れるので、影とならない部分を、角度 $\theta_1$ 、 $\theta_2$ として判別できる。センサs1、s2が固定されているため、角度 $\theta_1$ 、 $\theta_2$ から、操作物が検知面1100を横切る位置p(x, y)を特定できる。

検知面枠1000内に形成される検知面1100の各位置と、画面の各位置と、を一对一に対応付けておくことで、操作物が検知面1100を横切る位置に対応した画面上での位置を特定できる。

25

このようにして、操作物の画面上での位置あるいは位置の変化量を求めて、それを画面上のオブジェクトの動きに反映する。

しかしながら、上記のように、従来の画像生成システムでは、検知面枠1000を形成し、さらに、その両角部にセンサs1、s2を配置する必要がある。このため、システムが大掛かりなものとなって、高価になるばかりでなく、広い設置場所が必要となる。従って、この従来のシステムは、一般家庭に向いているとは言い難い。

- 5      また、検知面1100の各位置と、画面の各位置と、を一対一に対応付ける必要があるため、検知面枠1000の形状を決定する際の制約が大きい。このことも、設置場所が制限される一因となる。

さらに、オペレータは、検知面枠1000の範囲で、操作物を操作しなければならず、操作物を操作する際の制約が大きい。一方、操作物の操作の制約を小さくしようとすると、検知面枠1000を大きくしなければならず、設置場所の制限が大きくなるとともに、高価なものとなって、一般家庭での購入が困難となる。

- 10      さらに、二次元の検知面1100を横切るように、操作物を操作しなければならず、このこともまた、操作物を操作する際の制約を大きくする。つまり、二次元の検知面1100を横切らなければならないので、オペレータは、検知面1100に垂直なz軸方向に操作物を移動させることができなくなって、操作の自由度が小さくなる。上記文献に開示されているように、検知面枠を2つ設けても、この問題を
- 15      十分に解消することはできない。しかも、検知面枠を増やせば、上記した設置場所の問題及び価格の問題がさらに大きくなり、一般家庭での購入がさらに困難なものとなる。

#### 発明の開示

- そこで、本発明は、省スペースの実現及び操作の自由度の向上を図りながらも、オペレータにより操作される操作物の検出結果を反映した画像をディスプレイ装置に表示できる情報処理装置及びその関連技術を提供することを目的とする。

- 本発明の第1の形態によると、情報処理装置は、オペレータに保持されて動きが与えられる操作物の動きを加味した画像をディスプレイ装置に表示させる情報処理装置であって、反射面を有する前記操作物に、予め定められた周期で、光を照射するストロボスコープと、前記ストロボスコープの発光時及び
- 25      消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、前記差分信号を基に、前記操作物の状態情報を算出し、その状態情報に基づいて、第1のトリガを発生する状態情報算出ユニットと、前記第1のトリガに基づいて、前記操作物の移動軌跡を表現する第1のオブジェクトを前記ディスプレイ装置に表示させる画像表示処理ユニットと、を備える。

この構成によれば、ストロボスコープにより間欠的に光が照射される操作物を撮像ユニットにより撮影して、操作物の状態情報を求めている。このように、実空間に検知面（二次元）を形成することなく、撮像ユニットの撮影範囲である検知空間（三次元）に存在する操作物の状態情報を取得できる。従って、操作物の操作範囲が二次元平面に制限されることがないため、オペレータによる操作物の操作の制約が

5 小さく、操作の自由度を大きくできる。

また、ディスプレイ装置のスクリーンに対応した検知面を実空間に形成する必要もないため、設置場所の制限を小さくできる（省スペースの実現）。

さらに、操作物の状態情報に基づく第1のトリガにより、操作物の移動軌跡を表現する第1のオブジェクトがディスプレイ装置に表示される。このため、オペレータは、現実に見ることができない移動軌跡を、ディスプレイ装置上で見ることができ、操作物を操作した実感をより味合うことができる。

さらに、ディスプレイ装置に表示される仮想世界に、オペレータが操作した操作物の移動軌跡が出現する。このような操作物の移動軌跡の表示を通じて、オペレータは、仮想世界との接触が可能となり、仮想世界をより楽しむことができる。例えば、本発明の情報処理装置をゲーム装置として使用する場合は、オペレータは、あたかも、ディスプレイ装置に表示されるゲーム世界で、ゲームを楽しんでいるような実感を得ることができる。

さらに、発光時画像信号と消灯時画像信号との差分信号を生成するといった簡単な処理のみで、ノイズや外乱の影響を抑えた精度の高い検出が可能であるので、コスト、許容される消費電力等の条件により情報処理装置のパフォーマンスが制限されるシステムの上でも容易に達成が可能である。

ここで、本明細書において、操作物の「操作」とは、操作物を移動させること、又は、操作物を回転させること、等を意味し、スイッチを押下したり、アナログスティックを動かしたり、等は含まない。

上記の情報処理装置において、前記移動軌跡を表現する前記第1のオブジェクトは、帯状オブジェクトであり、前記画像表示処理ユニットは、フレームごとに幅の異なる前記帯状オブジェクトを前記ディスプレイ装置に表示することにより、前記操作物の移動軌跡を表現し、前記帯状オブジェクトの前記幅は、フレームの更新とともに太くなった後、フレームの更新とともに細くなる。

25 この構成によれば、鋭い閃光が走ったような移動軌跡を表示できる。特に、帯状オブジェクトの色を工夫することで、その効果をより高めることができる。

上記の情報処理装置において、前記画像表示処理ユニットは、第2のオブジェクトを前記ディスプレイ装置に表示し、前記状態情報算出ユニットは、前記第2のオブジェクトと、前記操作物の移動軌跡を表現する前記第1のオブジェクトと、の位置関係が所定条件を満たしたときに、第2のトリガを発生し、



前記画像表示処理ユニットは、前記第2のトリガに基づいて、予め定められたエフェクトを与えた前記第2のオブジェクトを前記ディスプレイ装置に表示する。

- 5 この構成によれば、位置関係が所定の条件を満たすように、オペレータが操作物を操作すると、その移動軌跡を表現する第1のオブジェクトを通じて、ディスプレイ装置に表示された、言わば仮想世界の第2のオブジェクトにエフェクトを与えることができる。このため、オペレータは、仮想世界をより一層楽しむことができる。

- 10 上記の情報処理装置において、前記状態情報算出ユニットは、前記操作物の前記状態情報としての速さ情報が、予め定められた第1の閾値を超えてから、予め定められた第2の閾値を下回るまでの、前記操作物の前記状態情報としての位置情報を算出し、あるいは、前記操作物の前記速さ情報が、前記予め定められた第1の閾値を超えてから、前記撮像ユニットによる撮影範囲を超える前までの、前記操作物の前記位置情報を算出し、かつ、前記操作物の前記位置情報の取得回数が3以上の場合に、前記操作物の最初の前記位置情報と、前記操作物の最後の前記位置情報と、に基づいて、前記操作物の移動軌跡を表現する前記第1のオブジェクトの態様を決定し、かつ、前記操作物の前記位置情報の取得回数が3以上の場合に、前記状態情報に基づいて、前記第1のトリガを発生する。

- 15 この構成によれば、操作物の位置情報の取得回数、つまり、操作物の検知回数が3以上の場合に、第1のトリガを発生するため、オペレータが意図しない操作により、意図しないときに、第1のオブジェクトが出現することを防止できる。

- 20 また、操作物の位置情報の取得回数（操作物の検知回数）が3以上の場合に、操作物の最初的位置情報と最後の位置情報とに基づいて、操作物の移動軌跡を表現する第1のオブジェクトの態様が決定される。このため、操作物の移動軌跡をより的確に反映した第1のオブジェクトの態様を決定できる。

- 25 なお、操作物の近接する2つの位置情報に基づいて、第1のオブジェクトの態様を決定すると、例えば、次の不都合がある。オペレータが、自分の感覚で直線的に操作物を移動させた場合でも、実際には、若干の円弧を描いている場合もある。この場合には、当然、撮像ユニットには、円弧を描くように操作物が撮影される。このときに、近接する2つの位置情報に基づいて、第1のオブジェクトの態様を決定すると、オペレータの感覚とずれた態様の第1のオブジェクトが表示されることになる。

ここで、第1のオブジェクトの態様とは、第1のオブジェクトの角度及び／又は方向等の、表示する第1のオブジェクトの形態のことである。

上記の情報処理装置において、前記状態情報算出ユニットは、前記操作物の前記状態情報としての面積情報を算出し、その面積情報が、予め定められた第3の閾値を超えたときに、第3のトリガを発生し、

前記画像表示処理ユニットは、前記第3のトリガに基づいて、第3のオブジェクトを前記ディスプレイ装置に表示する。

この構成によれば、操作物の広い反射面が撮影されたときに、第3のオブジェクトが表示されることになる。つまり、オペレータが、操作物の広い反射面を撮像ユニットに向けると、第3のオブジェクトが表示される。従って、単一の操作物の操作により、様々な画像を表示できる。また、様々な画像を表示するために、複数の操作物を用意したり、操作物にスイッチやアナログスティック等を設ける必要もなく、操作物の製造コストを低減できるばかりでなく、操作性をも向上できる。

上記の情報処理装置において、前記画像表示処理ユニットは、文字列を前記ディスプレイ装置に表示し、前記状態情報算出ユニットは、前記操作物の前記状態情報に基づいて、第4のトリガを発生し、前記画像表示処理ユニットは、前記第4のトリガに基づいて、前記文字列と異なる文字列を前記ディスプレイ装置に表示する。

この構成によれば、操作物の状態情報に基づいて、ディスプレイ装置上に文字列を次々に表示させることができるため、文字列の更新に使用するスイッチやアナログスティック等を操作物に設ける必要もなく、操作物の製造コストを低減できるばかりでなく、操作性をも向上できる。

上記の情報処理装置において、前記状態情報算出ユニットは、前記操作物の前記状態情報に基づいて、第5のトリガを発生し、前記画像表示処理ユニットは、前記第5のトリガに基づいて、背景画像を更新する。

この構成によれば、操作物の状態情報に基づいて、背景を更新することができるため、背景の更新に使用するスイッチやアナログスティック等を操作物に設ける必要もなく、操作物の製造コストを低減できるばかりでなく、操作性をも向上できる。

上記の情報処理装置において、前記操作物の前記状態情報としての位置情報を補正する補正情報を取得する補正情報取得ユニット、をさらに備え、前記状態情報算出ユニットは、前記補正情報を用いて、補正後の位置情報を算出する。

この構成によれば、オペレータが操作物を操作する感覚と、状態情報算出ユニットが算出する操作物の状態情報と、のずれを極力解消できるため、オペレータによる操作物の操作をよりの確に反映した画像を表示できる。

上記の情報処理装置において、前記画像表示処理ユニットは、カーソルを前記ディスプレイ装置に表示し、かつ、前記操作物の前記状態情報としての位置情報に応じて、前記カーソルを移動させる。

この構成によれば、操作物の状態情報に基づいて、カーソルを移動することができるため、カーソルの移動に使用するスイッチやアナログスティック等を操作物に設ける必要もなく、操作物の製造コストを低減できるばかりでなく、操作性をも向上できる。

- 上記の情報処理装置において、前記操作物の前記状態情報に基づいて、予め定められた処理を実行する  
5 ことを確定する。

この構成によれば、操作物の状態情報に基づいて、処理の実行の確定ができるため、処理の実行の確定に使用するスイッチやアナログスティック等を操作物に設ける必要もなく、操作物の製造コストを低減できるばかりでなく、操作性をも向上できる。

- 上記の情報処理装置において、前記画像表示処理ユニットは、前記カーソルが第4のオブジェクトに  
10 重ねて表示されたときに、その第4のオブジェクトに関連付けられた画像を前記ディスプレイ装置に表示する。

この構成によれば、オペレータは、操作物の操作によりカーソルを移動させるだけで、表示されている第4のオブジェクトに関連付けられた画像を表示させることができる。

- 上記の情報処理装置において、前記画像表示処理ユニットは、前記カーソルにより選択された文字を  
15 前記ディスプレイ装置に表示する。

この構成によれば、オペレータは、操作物の操作によりカーソルを移動させて所望の文字を選択するだけで、文字を入力できるため、文字入力に使用するスイッチやアナログスティック等を操作物に設ける必要もなく、操作物の製造コストを低減できるばかりでなく、操作性をも向上できる。

- 上記の情報処理装置において、前記状態情報算出ユニットは、前記操作物の前記状態情報に基づいて、  
20 第6のトリガを発生し、前記画像表示処理ユニットは、前記第6のトリガに基づいて、前記操作物の動きに応じた第5のオブジェクトを前記ディスプレイ装置に表示する。

この構成によれば、操作物の移動軌跡を表現する第1のオブジェクトと異なった視覚的效果を、オペレータに与えることができる。

- 上記の情報処理装置において、前記画像表示処理ユニットは、前記第1のトリガが発生してから、所  
25 定時間経過後に、前記操作物の移動軌跡を表現する前記第1のオブジェクトを前記ディスプレイ装置に表示する。

この構成によれば、操作物の移動軌跡を表現する第1のオブジェクトを、第1のトリガの発生とほぼ同時（人間の感覚で同時）に表示する場合と比較して、異なった効果を、オペレータに与えることができる。

上記の情報処理装置において、前記画像表示処理ユニットは、前記操作物の連続した前記状態情報が所定条件を満足したときに、第6のオブジェクトを表示する。

- この構成によれば、操作物の操作が所定条件を満足したときにのみ、第6のオブジェクトが表示されるため、この所定条件の設定の仕方によって、第6のオブジェクトを表示させるための、オペレータによる操作物の操作を任意にコントロールできる。

上記の情報処理装置において、前記画像表示処理ユニットは、前記操作物の操作方向及び操作タイミングを指示するガイドを表示する。

この構成によれば、オペレータは、情報処理装置が要求する操作物の操作方向及び操作タイミングを視覚的に認識できる。

- 10 上記の情報処理装置において、前記操作物の前記状態情報は、速さ情報、移動方向情報、移動距離情報、速度ベクトル情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、又は、それらの2以上の組み合わせ、である。

この構成によれば、オペレータによる操作物の様々な動きに応じたオブジェクトをディスプレイ装置に表示できる。

- 15 上記の情報処理装置において、前記第1のトリガに基づいて、スピーカから効果音を発生させる効果音生成ユニット、をさらに備える。

この構成によれば、オペレータに対して、視覚的效果に加えて、聴覚的效果を与えることができる。従って、オペレータは、ディスプレイ装置に表示される仮想世界をより一層楽しむことができる。例えば、オペレータが操作した操作物の移動軌跡が仮想世界に出現すると同時に、効果音を発生させれば、

- 20 オペレータは、仮想世界をより一層楽しむことができる。

本発明の第2の形態によると、情報処理装置は、オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置に画像を表示させる情報処理装置であって、複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するストロボスコープと、前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、前記差分信号を基に、前記操作物の状態情報を算出して、その状態情報に基づいて、前記複数の反射面のいずれの反射面が撮影されたかを判別する状態情報算出ユニットと、判別された前記反射面に応じて、前記ディスプレイ装置に異なる画像を表示させる画像表示処理ユニットと、を備える。

この構成によれば、ストロボスコープにより間欠的に光が照射される操作物を撮像ユニットにより撮影して、操作物の状態情報を求めている。このように、実空間に検知面（二次元）を形成することなく、撮像ユニットの撮影範囲である検知空間（三次元）に存在する操作物の状態情報を取得できる。従って、操作物の操作範囲が二次元平面に制限されることがないため、オペレータによる操作物の操作の制約が

5 小さく、操作の自由度を大きくできる。

また、ディスプレイ装置のスクリーンに対応した検知面を実空間に形成する必要もないため、設置場所の制限を小さくできる（省スペースの実現）。

さらに、検知された反射面に応じて、異なる画像が表示されるため、単一の操作物を操作するだけで、反射面の数に応じた、異なる画像を表示できる。このため、異なる画像ごとに、対応する操作物を用意

10 したり、スイッチやアナログスティック等を操作物に設けたり、する必要がない。従って、操作物のコストの低減が可能になるとともに、オペレータによる操作物の操作性を向上できる。

さらに、オペレータは、操作物のどの反射面を撮像ユニットに向けるかによって、所望の画像を表示できる。例えば、本発明の情報処理装置をゲーム装置として使用する場合は、オペレータは、単一の操作物で、様々な画像を表示させることができ、ゲームを円滑に実行できる。

15 さらに、発光時画像信号と消灯時画像信号との差分信号を生成するといった簡単な処理のみで、ノイズや外乱の影響を抑えた精度の高い検出が可能であるので、コスト、許容される消費電力等の条件により情報処理装置のパフォーマンスが制限されるシステムの上でも容易に達成が可能である。

上記の情報処理装置において、前記状態情報は、前記反射面の面積情報、数情報、形状情報、若しくは、形状を表す比率情報、のいずれか、又は、それらの組み合わせ、である。

20 この構成によれば、状態情報算出ユニットは、これらの情報により、操作物のいずれの反射面が撮影されたのかを判別することができる。従って、反射面のサイズ又は形状を異ならせるだけで、いずれの反射面が撮影されたのかを容易に判別できる。特に、面積情報により、反射面を判別する場合は、過った判別を極力減少させることができるだけでなく、処理を容易にできて処理の高速化を図ることができる。

25 本発明の第3の形態によると、情報処理装置は、オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置に画像を表示させる情報処理装置であって、複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するストロボスコープと、前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニッ

トと、前記差分信号に基づいて、前記反射面ごとの状態情報を算出する状態情報算出ユニットと、前記複数の反射面の前記状態情報に応じて、画像を表示させる画像表示処理ユニットと、を備える。

この構成によれば、ストロボスコープにより間欠的に光が照射される操作物を撮像ユニットにより撮影して、操作物の状態情報を求めている。このように、実空間に検知面（二次元）を形成することなく、

- 5 撮像ユニットの撮影範囲である検知空間（三次元）に存在する操作物の状態情報を取得できる。従って、操作物の操作範囲が二次元平面に制限されることがないため、オペレータによる操作物の操作の制約が小さく、操作の自由度を大きくできる。

また、ディスプレイ装置のスクリーンに対応した検知面を実空間に形成する必要もないため、設置場所の制限を小さくできる（省スペースの実現）。

- 10 さらに、複数の反射面の状態情報に応じて画像を表示するため、単一の反射面の状態情報に応じて画像を表示する場合と比較して、操作物の状態をより反映した画像を表示できる。

さらに、発光時画像信号と消灯時画像信号との差分信号を生成するといった簡単な処理のみで、ノイズや外乱の影響を抑えた精度の高い検出が可能であるので、コスト、許容される消費電力等の条件により情報処理装置のパフォーマンスが制限されるシステムの上でも容易に達成が可能である。

- 15 本発明の第4の形態によると、ゲームシステムは、ゲームを行うためのゲームシステムであって、オペレータによって実際に動かされる操作物と、オペレータによって動かされている操作物を撮影するイメージセンサと、ゲームを行う際にディスプレイ装置に接続され、前記イメージセンサからイメージ信号を受け取り、前記ゲームのコンテンツを前記ディスプレイ装置に表示する処理装置と、を備え、前記操作物は、前記イメージセンサによって撮影された前記操作物のイメージに基づいて、前記ゲームにおいて所定の役割を担っており、前記ゲームを行っている際に、前記処理装置によって前記ディスプレイ装置に表示される前記ゲームのコンテンツにおける、前記操作物の移動軌跡の表示は、帯状イメージに簡略化されており、この帯状イメージは、前記オペレータによって動かされる前記操作物の、前記ディスプレイ装置上の表示における移動軌跡の少なくとも2点を結ぶものであり、この少なくとも2点は、前記イメージセンサで撮影されたイメージから取得される。

- 25 本発明の新規な特徴は、特許請求の範囲に記載されている。しかしながら、発明そのもの及びその他の特徴と効果は、添付図面を参照して具体的な実施例の詳細な説明を読むことにより容易に理解される。

#### 図面の簡単な説明

図1は、本発明の実施の形態における情報処理システムの全体構成を示す図である。

図2は、図1の情報処理装置及び剣の拡大図である。

図3は、図2の剣の上面図である。

図4は、図1の剣の他の例の拡大図である。

図5は、図4の剣の上面図である。

5 図6は、図2の撮像ユニットの一例を示す図解図である。

図7は、図1の情報処理装置の電氣的な構成を示す図である。

図8は、図7の高速プロセッサのブロック図である。

図9は、図7のイメージセンサから高速プロセッサへ 픽セルデータを取り込む構成及びLED駆動回路を示す回路図である。

10 図10(a)は、図9のイメージセンサが出力するフレームステータスフラグ信号FSFのタイミング図である。図10(b)は、図9のイメージセンサが出力する 픽セルデータストローブ信号PDSのタイミング図である。図10(c)は、図9のイメージセンサが出力する 픽セルデータD(X, Y)のタイミング図である。図10(d)は、図9の高速プロセッサが出力するLEDコントロール信号LEDLのタイミング図である。図10(e)は、図9の赤外発光ダイオードの点灯状態を示すタイミング図である。図10(f)は、図9のイメージセンサの露光期間を示すタイミング図である。

図11(a)は、図10のフレームステータスフラグ信号FSFの拡大図である。図11(b)は、図10の 픽セルデータストローブ信号PDSの拡大図である。図11(c)は、図10の 픽セルデータD(X, Y)の拡大図である。

図12は、図1のテレビジョンモニタのスクリーンに表示される選択画面の例示図である。

20 図13は、図12の選択画面で「ストーリーモード」の項目オブジェクトが選択されたときのゲーム画面の例示図である。

図14は、図12の選択画面で「ストーリーモード」の項目オブジェクトが選択されたときのゲーム画面の他の例示図である。

25 図15は、図12の選択画面で「ストーリーモード」の項目オブジェクトが選択されたときのゲーム画面のさらに他の例示図である。

図16は、図12の選択画面で「ストーリーモード」の項目オブジェクトが選択されたときのゲーム画面のさらに他の例示図である。

図17は、図12の選択画面で「ストーリーモード」の項目オブジェクトが選択されたときのゲーム画面のさらに他の例示図である。

図18 (a) は、図12の選択画面で「ストーリーモード」の項目オブジェクトが選択されたときのゲーム画面のさらに他の例示図である。図18 (b) は、図18 (a) の更新後のゲーム画面の例示図である。

図19は、図12の選択画面で「対戦モード」の項目オブジェクトが選択されたときのゲーム画面の例示図である。

図20は、図7のROMに格納されたプログラム及びデータを示す概念図である。

図21 (a) は、一般的なイメージセンサにより撮影された、特別な処理を施さない画像の例示図である。図21 (b) は、図21 (a) の画像信号を或る閾値でレベル弁別したときの画像信号の例示図である。図21 (c) は、赤外フィルタを介したイメージセンサの点灯時の画像信号を或る閾値でレベル弁別したときの画像信号の例示図である。図21 (d) は、赤外フィルタを介したイメージセンサの消灯時の画像信号を或る閾値でレベル弁別したときの画像信号の例示図である。図21 (e) は、点灯時の画像信号と消灯時の画像信号との差分信号の例示図である。

図22は、図7の高速プロセッサが剣のスイングを検出する際の説明図である。

図23 (a) は、実施の形態における角度フラグの値と角度との関係図である。図23 (b) は、実施の形態における方向フラグの値と方向を表す符号との関係図である。図23 (c) は、実施の形態における角度フラグ及び方向フラグと、スイング情報と、の関係図である。

図24は、図23 (c) のスイング情報と、剣の操作方向と、の関係図である。

図25は、図23 (c) のスイング情報とアニメーションテーブル格納位置情報との関係図である。

図26は、図7のROMに格納された、剣軌跡オブジェクトをアニメーションするためのアニメーションテーブルの例示図である。

図27は、図14の剣軌跡オブジェクトのアニメーションを行うためのオブジェクト画像データの例示図である。

図28は、図14の剣軌跡オブジェクトのアニメーションを行うためのオブジェクト画像データの他の例示図である。

図29は、図14の剣軌跡オブジェクトのアニメーションを行うためのオブジェクト画像データの他の例示図である。

図30は、図7の高速プロセッサによるヒット判定の説明図である。

図31は、図12の選択画面で「スイング補正」の項目オブジェクトが選択されたときの、スイング補正画面の例示図である。



- 図32は、図1の情報処理装置の全体の処理の流れを示すフローチャートである。
- 図33は、図32のステップS1の初期設定処理の流れを示すフローチャートである。
- 図34は、図33のステップS20のセンサ初期設定処理の流れを示すフローチャートである。
- 図35は、図34のステップS31のコマンド送信処理の流れを示すフローチャートである。
- 5 図36(a)は、図9のレジスタ設定クロックCLKのタイミング図である。図36(b)は、図9のレジスタデータのタイミング図である。
- 図37は、図34のステップS33のレジスタ設定処理の流れを示すフローチャートである。
- 図38は、図32のステップS7のストーリーモードの流れを示すフローチャートである。
- 図39は、図38のステップS60のピクセルデータ群取得処理の流れを示すフローチャートである。
- 10 図40は、図39のステップS81のピクセルデータ取得処理の流れを示すフローチャートである。
- 図41は、図38のステップS61の注目領域抽出処理の流れを示すフローチャートである。
- 図42は、図38のステップS62の注目点抽出処理の流れを示すフローチャートである。
- 図43は、図38のステップS63のスイング検出処理の流れを示すフローチャートである。
- 図44は、図43のステップS166の剣軌跡種類決定処理の流れを示すフローチャートである。
- 15 図45は、図43のステップS167の剣軌跡座標算出処理の流れを示すフローチャートである。
- 図46は、図38のステップS64のヒット判定処理の流れを示すフローチャートである。
- 図47は、図38のステップS65の盾検出処理の流れを示すフローチャートである。
- 図48は、図38のステップS66の説明進行処理の流れを示すフローチャートである。
- 図49は、図38のステップS67の前進処理の流れを示すフローチャートである。
- 20 図50は、図38のステップS70の画像表示処理の流れを示すフローチャートである。
- 図51は、図32のステップS5のモード選択処理の流れを示すフローチャートである。
- 図52は、図51のステップS303のカーソル移動処理の流れを示すフローチャートである。
- 図53は、図51のステップS304の項目オブジェクト移動処理の流れを示すフローチャートである。
- 図54は、図32のステップS6のスイング補正モードの流れを示すフローチャートである。
- 25 図55は、図54のステップS404の補正情報取得処理の流れを示すフローチャートである。
- 図56は、図6の撮像ユニットによるストロボスコープ撮影処理の流れを示すフローチャートである。
- 図57は、本実施の形態におけるゲーム画面の他の例示図である。
- 図58は、本実施の形態におけるゲーム画面のさらに他の例示図である。
- 図59は、本実施の形態におけるゲーム画面のさらに他の例示図である。

図60は、本実施の形態におけるゲーム画面のさらに他の例示図である。

図61(a)は、図1の剣のさらに他の例示図である。図61(b)は、図1の剣のさらに他の例示図である。図61(c)は、図1の剣のさらに他の例示図である。

図62は、実施の形態における操作物の他の例示図である。

5 図63は、実施の形態における第1の反射シートの注目点の座標算出の説明図である。

図64は、実施の形態における第2の反射シートの注目点の座標算出の説明図である。

図65は、従来の画像生成システムの説明図である。

#### 発明を実施するための最良の形態

10 以下、本発明の実施の形態について、図面を参照しながら説明する。なお、図中、同一または相当部分については同一の参照符号を付してその説明を援用する。

図1は、本発明の実施の形態における情報処理システムの全体構成を示す図である。図1に示すように、この情報処理システムは、情報処理装置1、操作物3、及び、テレビジョンモニタ90、を含む。

本実施の形態では、操作物3の1例として、剣型の操作物3（以下、本実施の形態では、「剣3」と呼ぶ。）を挙げている。そして、本実施の形態では、情報処理の1例として、ゲーム処理を挙げる。

情報処理装置1には、ACアダプタ92により、直流電源電圧が与えられる。ただし、ACアダプタ92に代えて、電池（図示せず）により、直流電源電圧を与えることもできる。

テレビジョンモニタ90には、その前面にスクリーン91が設けられる。テレビジョンモニタ90と情報処理装置1とは、AVケーブル93により接続される。

20 情報処理装置1は、例えば、図1に示すように、テレビジョンモニタ90の上面に載置される。

図2は、図1の情報処理装置1及び剣3の拡大図である。図3は、図2の剣3の上面図である。

図2に示すように、情報処理装置1のハウジング11には、撮像ユニット5が組み込まれる。撮像ユニット5は、4つの赤外発光ダイオード7及び赤外フィルタ9を含む。赤外発光ダイオード7の発光部は、赤外フィルタ9から露出している。

25 撮像ユニット5の赤外発光ダイオード7は、間欠的に赤外光を発光する。赤外発光ダイオード7からの赤外光は、剣3により反射され、赤外フィルタ9の内部に設けられた撮像素子（後述）に入力される。このようにして、剣3が間欠的に撮影される。従って、情報処理装置1は、オペレータ94により振り動かされた剣3の間欠的な画像信号を取得できる。情報処理装置1は、この画像信号を解析して、その解析結果を、ゲーム処理に反映する。

また、情報処理装置1の背面から、メモリカートリッジ13を装着することができる。このメモリカートリッジ13には、EEPROM (electrically erasable and programmable read only memory) が内蔵されている(図示せず)。このEEPROMには、1人でプレイするストーリーモードのゲーム結果を保存することができる。

- 5      また、図2及び図3に示すように、剣3の刀身部15の両面には、反射シート17が取り付けられる。この反射シート17により、反射面が形成される。また、剣3の鍔部19の両面には、半円柱状部材21が取り付けられる。この半円柱状部材21の曲面には、反射シート23が取り付けられる。この反射シート23により、反射面が形成される。反射シート17, 23は、例えば、再帰反射シートである。

- 10      また、図2に示すように、剣3の柄頭25には、ストラップ27が取り付けられる。オペレータ94は、このストラップ27を手首に通して、剣3の柄29を握る。これにより、オペレータ94が、過って柄29から手を離れた場合でも、剣3が予期せぬ所へ飛んでいくの防止し、安全が図られる。

- 15      図4は、図1の剣3の他の例の拡大図である。図5は、図4の剣3の上面図である。図4及び図5の剣3には、図2及び図3の半円柱状部材21が設けられていない。その代わり、図4及び図5の剣3では、その刃先部に反射シート31(例えば、再帰反射シート)が取り付けられる。図4及び図5の剣3では、反射シート31が、図2及び図3の剣3の反射シート23の機能を果たすことになる。なお、以下では、図2及び図3に示した剣3を用いて説明を行う。

- 20      図6は、図2の撮像ユニット5の一例を示す図解図である。図6に示すように、この撮像ユニット5は、たとえばプラスチック成型によって形成されるユニットベース45を含み、このユニットベース45内には支持筒47が取り付けられる。支持筒47の上面には内面が逆円錐形状であるラッパ状の開口41が形成され、その開口41の下方の筒状部内部には、いずれもがたとえば透明プラスチックの成型によって形成された凹レンズ49および凸レンズ51を含む光学系が設けられ、凸レンズ51の下方において、撮像素子としてのイメージセンサ43が固着される。したがって、イメージセンサ43は、開口41からレンズ49および51を通して入射する光に応じた画像を撮影することができる。

- 25      イメージセンサ43は、低解像度のCMOSイメージセンサ(たとえば32ピクセル×32ピクセル: グレースケール)である。ただし、このイメージセンサ43は、画素数のもっと多いものでもよいし、CCD等の他の素子からなるものであってもよい。以下では、イメージセンサ43が、32ピクセル×32ピクセルからなるものとして説明を行う。

また、ユニットベース45には、光出射方向がいずれも上方向とされた複数(実施の形態では4つ)の赤外発光ダイオード7が取り付けられる。この赤外発光ダイオード7によって、撮像ユニット5の上

方に赤外光が照射される。また、ユニットベース45の上方には、赤外フィルタ（赤外光のみを透過するフィルタ）9が上記開口41を覆うように、取り付けられる。そして、赤外発光ダイオード7は後述のように、点灯／消灯（非点灯）が連続的に繰り返されるので、ストロボスコープとして機能する。ただし、「ストロボスコープ」とは、運動体を間欠的に照らす装置の総称である。したがって、上記イメージセンサ43は、その撮影範囲内で移動する物体、実施の形態では、剣3を撮影することになる。なお、後述する図9に示すように、ストロボスコープは、主に、赤外発光ダイオード7、LED駆動回路82、及び、高速プロセッサ200、により構成される。

ここで、撮像ユニット5は、イメージセンサ43の受光面が、水平面から所定角度（例えば、90度）だけ傾くように、ハウジング11に組み込まれる。また、凹レンズ49および凸レンズ51により、イメージセンサ43の撮影範囲は、例えば、60度の範囲である。

図7は、図1の情報処理装置1の電気的な構成を示す図である。図7に示すように、情報処理装置1は、イメージセンサ43、赤外発光ダイオード7、映像信号出力端子61、音声信号出力端子63、高速プロセッサ200、ROM（read only memory）65、及び、バス67、を含む。

高速プロセッサ200には、バス67が接続される。さらに、バス67には、ROM65が接続される。従って、高速プロセッサ200は、バス67を介して、ROM65にアクセスすることができるので、ROM65に格納された制御プログラムをリードして実行でき、また、ROM65に格納された画像データ及び音声データをリードして処理し、映像信号及び音声信号を生成して、映像信号出力端子61及び音声信号出力端子63に出力することができる。

また、情報処理装置1の背面にはメモ리카ートリッジ13を装着するためのコネクタ（図示せず）が設けられる。従って、高速プロセッサ200は、バス67を介して、そのコネクタに装着されたメモ리카ートリッジ13に内蔵されるEEPROM69にアクセスできる。これにより、高速プロセッサ200は、バス67を介して、EEPROM69に格納されたデータをリードして、ゲーム処理に利用することができる。

さて、剣3は、赤外発光ダイオード7が発光する赤外光に照射され、その赤外光を反射シート17, 23で反射する。この反射シート17, 23からの反射光がイメージセンサ43によって検知され、したがって、イメージセンサ43からは反射シート17, 23の画像信号が出力される。イメージセンサ43からのこのアナログ画像信号は高速プロセッサ200に内蔵されたA/Dコンバータ（後述）によってデジタルデータに変換される。そして、高速プロセッサ200は、このデジタルデータを解析して、その解析結果をゲーム処理に反映する。

図8は、図7の高速プロセッサ200のブロック図である。図8に示すように、この高速プロセッサ200は、中央演算処理装置（CPU：central processing unit）201、グラフィックプロセッサ202、サウンドプロセッサ203、DMA（direct memory access）コントローラ204、第1バス調停回路205、第2バス調停回路206、内部メモリ207、A/Dコンバータ（ADC：analog to digital converter）208、入出力制御回路209、タイマ回路210、DRAM（dynamic random access memory）リフレッシュ制御回路211、外部メモリインタフェース回路212、クロックドライバ213、PLL（phase-locked loop）回路214、低電圧検出回路215、第1バス218、及び、第2バス219、を含む。

- 10 CPU201は、メモリ（内部メモリ207、又は、ROM65）に格納されたプログラムに従い、各種演算やシステム全体の制御を行う。CPU201は、第1バス218及び第2バス219のバスマスタであり、それぞれのバスに接続された資源にアクセスが可能である。

- 15 グラフィックプロセッサ202は、第1バス218及び第2バス219のバスマスタであり、内部メモリ207、又は、ROM65に格納されたデータを基に、映像信号を生成して、映像信号出力端子61へ出力する。グラフィックプロセッサ202は、第1バス218を通じて、CPU201により制御される。また、グラフィックプロセッサ202は、CPU201に対して、割込み要求信号220を発生する機能を有する。

- 20 サウンドプロセッサ203は、第1バス218及び第2バス219のバスマスタであり、内部メモリ207、又は、ROM65に格納されたデータを基に、音声信号を生成して、音声信号出力端子63へ出力する。サウンドプロセッサ203は、第1バス218を通じて、CPU201により制御される。また、サウンドプロセッサ203は、CPU201に対して、割込み要求信号220を発生する機能を有する。

- 25 DMAコントローラ204は、ROM65及びEEPROM69から、内部メモリ207へのデータ転送を司る。また、DMAコントローラ204は、データ転送の完了を通知するために、CPU201に対する割込み要求信号220を発生する機能を有する。DMAコントローラ204は、第1バス218及び第2バス219のバスマスタである。DMAコントローラ204は、第1バス218を通じてCPU201により制御される。

第1バス調停回路205は、第1バス218の各バスマスタからの第1バス使用要求信号を受け付け、調停を行い、各バスマスタへの第1バス使用許可信号を発行する。各バスマスタは、第1バス使用許可

信号を受領することによって第1バス218に対するアクセスが許可される。ここで、第1バス使用要求信号及び第1バス使用許可信号は、図8では、第1バス調停信号222として示されている。

第2バス調停回路206は、第2バス219の各バスマスタからの第2バス使用要求信号を受け付け、調停を行い、各バスマスタへの第2バス使用許可信号を発行する。各バスマスタは、第2バス使用許可信号を受領することによって第2バス219に対するアクセスが許可される。ここで、第2バス使用要求信号及び第2バス使用許可信号は、図8では、第2バス調停信号223として示されている。

内部メモリ207は、マスクROM、SRAM (static random access memory)、及び、DRAMのうち、必要なものを備える。バッテリーによるSRAMのデータ保持が必要とされる場合、バッテリー217が必要となる。DRAMが搭載される場合、定期的にリフレッシュと呼ばれる記憶内容保持のための動作が必要とされる。

ADC208は、アナログ入力信号をデジタル信号に変換する。このデジタル信号は、第1バス218を介してCPU201によってリードされる。また、ADC208は、CPU201に対して、割込み要求信号220を発生する機能を有する。

このADC208が、イメージセンサ43からのピクセルデータ（アナログ）を受けて、デジタルデータに変換する。

入出力制御回路209は、外部入出力装置や外部の半導体素子との通信等を入出力信号を介して行う。入出力信号は、第1バス218を介して、CPU201からリード／ライトされる。また、入出力制御回路209は、CPU201に対して、割込み要求信号220を発生する機能を有する。

この入出力制御回路209から、赤外発光ダイオード7を制御するLEDコントロール信号LEDCが出力される。

タイマ回路210は、設定された時間間隔に基づき、CPU201に対する割込み要求信号220を発生する機能を有する。時間間隔等の設定は、第1バス218を介してCPU201によって行われる。

DRAMリフレッシュ制御回路211は、一定期間毎に第1バス218の使用権を無条件に獲得し、DRAMのリフレッシュ動作を行う。なお、DRAMリフレッシュ制御回路211は、内部メモリ207がDRAMを含む場合に設けられる。

PLL回路214は、水晶振動子216より得られる正弦波信号を逡倍した高周波クロック信号を生成する。

クロックドライバ213は、PLL回路214より受け取った高周波クロック信号を、各ブロックへクロック信号225を供給するのに十分な信号強度へ増幅する。

低電圧検出回路215は、電源電圧 $V_{cc}$ を監視し、電源電圧 $V_{cc}$ が一定電圧以下のときに、PLL回路214のリセット信号226、その他のシステム全体のリセット信号227を発行する。また、内部メモリ207がSRAMで構成されており、かつ、SRAMのバッテリ217によるデータ保持が要求される場合、電源電圧 $V_{cc}$ が一定電圧以下のときに、バッテリバックアップ制御信号224を発

5 行する機能を有する。

外部メモリインタフェース回路212は、第2バス219をバス67に接続するための機能を有する。

ここで、図9～図11を参照して、イメージセンサ43から高速プロセッサ200へピクセルデータを取り込むための構成を詳細に説明する。

図9は、図7のイメージセンサ43から高速プロセッサ200へピクセルデータを取り込む構成及び  
10 LED駆動回路を示す回路図である。図10は、図7のイメージセンサ43から高速プロセッサ200へピクセルデータを取り込む際の動作を示すタイミング図である。図11は、図10の一部を拡大して示すタイミング図である。

図9に示すように、イメージセンサ43は、ピクセルデータD(X, Y)をアナログ信号として出力するタイプのものであるため、このピクセルデータD(X, Y)は高速プロセッサ200のアナログ入  
15 力ポートに入力される。アナログ入力ポートは、この高速プロセッサ200においてADC208に接続され、したがって、高速プロセッサ200は、ADC208からデジタルデータに変換されたピクセルデータをその内部に取得する。

上述のアナログピクセルデータD(X, Y)の中点は、イメージセンサ43の基準電圧端子 $V_{ref}$ に与えられる基準電圧によって決定される。そのため、イメージセンサ43に関連して例えば抵抗分圧  
20 回路からなる基準電圧発生回路81が設けられ、この回路81から基準電圧端子 $V_{ref}$ に常に一定の大きさの基準電圧が与えられる。

イメージセンサ43を制御するための各デジタル信号は、高速プロセッサ200のI/Oポートに与えられ、またはそこから出力される。このI/Oポートは各々入力/出力の制御が可能なデジタルポートであり、この高速プロセッサ200で入出力制御回路209に接続されている。

25 詳しく言うと、高速プロセッサ200の出力ポートからはイメージセンサ43をリセットするためのリセット信号 $reset$ が出力され、イメージセンサ43に与えられる。また、イメージセンサ43からは、ピクセルデータストローブ信号PDSおよびフレームステータスフラグ信号FSFが出力され、それらの信号が高速プロセッサ200の入力ポートに与えられる。

ピクセルデータストローブ信号PDSは上述の各ピクセルデータD(X, Y)を読み込むための図10(b)に示すようなストローブ信号である。フレームステータスフラグ信号FSFはイメージセンサ43の状態を示すフラグ信号で、図10(a)に示すように、このイメージセンサ43の露光期間を規定する。つまり、フレームステータスフラグ信号FSFの図10(a)に示すローレベルが露光期間を示し、図10(a)に示すハイレベルが非露光期間を示す。

また、高速プロセッサ200は、イメージセンサ43の制御レジスタ(図示せず)に設定するコマンド(またはコマンド+データ)をレジスタデータとしてI/Oポートから出力するとともに、たとえばハイレベルおよびローレベルを繰り返すレジスタ設定クロックCLKを出力し、それらをイメージセンサ43に与える。

10     なお、赤外発光ダイオードとして、図9に示すように互いに並列接続された4つの赤外発光ダイオード7a, 7b, 7cおよび7dを用いる。この4つの赤外発光ダイオード7a~7dは、上で説明したように、剣3を照らすように、イメージセンサ43の視点方向と同一方向に赤外光を照射するようにかつイメージセンサ43を囲むように配置される。ただし、これら個別の赤外発光ダイオード7a~7dは、特に区別する必要がある場合を除いて、単に赤外発光ダイオード7と呼ばれる。

15     この赤外発光ダイオード7はLED駆動回路82によって、点灯されまたは消灯(非点灯)される。LED駆動回路82は、イメージセンサ43から上述のフレームステータスフラグ信号FSFを受け、このフラグ信号FSFは、抵抗83およびコンデンサ84からなる微分回路85を通してPNPトランジスタ86のベースに与えられる。このPNPトランジスタ86にはさらにプルアップ抵抗87が接続されていて、このPNPトランジスタ86のベースは、通常は、ハイレベルにプルアップされている。

20     そして、フレームステータス信号FSFがローレベルになると、そのローレベルが微分回路85を経てベースに入力されるため、PNPトランジスタ86は、フラグ信号FSFがローレベル期間にのみオンする。

PNPトランジスタ86のエミッタは抵抗88および89を介して接地される。そして、エミッタ抵抗88および89の接続点がNPNトランジスタ31のベースに接続される。このNPNトランジスタ31のコレクタが各赤外発光ダイオード7a~7dのアノードに共通に接続される。NPNトランジスタ31のエミッタが別のNPNトランジスタ33のベースに直接接続される。NPNトランジスタ33のコレクタが各赤外発光ダイオード7a~7dのカソードに共通接続され、エミッタが接地される。



このLED駆動回路82では、高速プロセッサ200のI/Oポートから出力されるLEDコントロール信号LEDCがアクティブ（ハイレベル）でありかつイメージセンサ43からのフレームステータスフラグ信号FSFがローレベルである期間にのみ赤外発光ダイオード7が点灯される。

- 図10（a）に示すようにフレームステータスフラグ信号FSFがローレベルになると、そのローレベル期間中（実際には微分回路85の時定数での遅れがあるが）、PNPトランジスタ86がオンする。したがって、図10（d）に示すLEDコントロール信号LEDCが高速プロセッサ200からハイレベルで出力されると、NPNトランジスタ31のベースがハイレベルとなり、このトランジスタ31がオンとなる。トランジスタ31がオンするとトランジスタ33はオンとなる。したがって、電源（図9では小さい白丸で示す）から各赤外発光ダイオード7a～7dおよびトランジスタ33を経て電流が流れ、応じて図10（e）に示すように各赤外発光ダイオード7a～7dが点灯される。

LED駆動回路82では、このように、図10（d）のLEDコントロール信号LEDCがアクティブでありかつ図10（a）のフレームステータスフラグ信号FSFがローレベルである期間にのみ赤外発光ダイオード7が点灯されるので、イメージセンサ43の露光期間（図10（f）参照）にのみ赤外発光ダイオード7が点灯されることになる。

- したがって、無駄な電力消費を抑制することができる。さらに、フレームステータスフラグ信号FSFはコンデンサ84によってカップリングされているので、万一イメージセンサ43の暴走等によりそのフラグ信号FSFがローレベルのまま停止した場合でも、一定時間後にはトランジスタ86は必ずオフされ、赤外発光ダイオード7も一定時間後には必ずオフされる。

- このように、フレームステータス信号FSFの持続期間を変更することによって、イメージセンサ43の露光時間を任意にかつ自在に設定または変更することができる。

さらに、フレームステータス信号FSFおよびLEDコントロール信号LEDCの持続時間や周期を変更することによって、赤外発光ダイオード7すなわちストロボ스코プの発光期間、非発光期間、発光／非発光周期などを任意にかつ自在に変更または設定できる。

- 先に説明したように、赤外発光ダイオード7からの赤外光によって剣3が照射されると、剣3からの反射光によってイメージセンサ43が露光される。応じて、イメージセンサ43から上述のピクセルデータD（X，Y）が出力される。詳しく説明すると、イメージセンサ43は、上述の図10（a）のフレームステータスフラグ信号FSFがハイレベルの期間（赤外発光ダイオード7の非点灯期間）に図10（b）に示すピクセルデータストローブPDSに同期して、図10（c）に示すようにアナログのピクセルデータD（X，Y）を出力する。

高速プロセッサ200では、そのフレームステータスフラグ信号FSFとピクセルデータストローブPDSとを監視しながら、ADC208を通じて、デジタルのピクセルデータを取得する。

ただし、ピクセルデータは、図11(c)に示すように、第0行、第1行、…第31行と行順次に出  
力される。ただし、後に説明するように、各行の先頭の1ピクセルはダミーデータとなる。ここで、イ  
メージセンサ43の水平方向（横方向、行方向）をX軸とし、垂直方向（縦方向、列方向）をY軸とし  
て、原点を左上の角とする。

さて、次に、情報処理装置1によるゲーム処理について、具体例を挙げながら説明する。

図12は、図1のテレビジョンモニタ90のスクリーン91に表示される選択画面の例示図である。  
オペレータ94が、情報処理装置1の背面に設けられた電源スイッチ（図示せず）をオンにすると、例  
えば、図12に示すような選択画面が表示される。本実施の形態では、選択できる項目の1例として、  
「ストーリーモードA」～「ストーリーモードE」（包括的に表すときは、「ストーリーモード」と表  
記する。）、「対戦モード」、及び、「スイング補正モード」、を挙げている。

選択画面には、剣型のカーソル101、左移動指示オブジェクト103、右移動指示オブジェクト1  
05、選択棒107、及び、項目オブジェクト109、が表示される。オペレータ94が、剣3を動か  
すと、その動きに応じて、画面上のカーソル101が移動する。このカーソル101を、左移動指示オ  
ブジェクト103に重ねると、項目オブジェクト109が、左方向に移動する。同様に、このカーソル  
101を、右移動指示オブジェクト105に重ねると、項目オブジェクト109が、右方向に移動する。

このように、オペレータ94は、剣3によりカーソル101を操作して、選択したい項目オブジェク  
ト109を、選択棒107内に静止させる。オペレータ94が、剣3を一定速度以上の大きさと振り下  
ろすと、選択が確定する。すると、情報処理装置1は、選択が確定した項目オブジェクト109に対応  
した処理を実行する。以下、オペレータ94が選択可能な各項目での処理について、図面を用いて説明  
する。

図13～図18(a)(b)は、図12の選択画面で「ストーリーモード」の項目オブジェクト10  
9が選択されたときのゲーム画面の例示図である。ストーリーモードでは、スクリーン91には、図1  
3示すようなゲーム画面が表示され、オペレータ94が1人で行うゲーム処理が実行される。このゲー  
ム画面には、ゲームストーリーに沿って、敵オブジェクト115が表示される。

また、オペレータ94が剣3を横方向（水平方向）に振ると、剣3を振ったことがトリガとなって、  
ゲーム画面には、図14に示すような横方向の剣軌跡オブジェクト117が出現する。剣軌跡オブジェ  
クト117は、剣3の実空間での移動軌跡（斬り跡）を表現するオブジェクトである。従って、図示は

省略しているが、剣3が斜めに振られると、斜め方向に、剣軌跡オブジェクト117が出現し、剣3が縦方向（垂直方向）に振られると、縦方向に、剣軌跡オブジェクト117が出現する。

このような剣軌跡オブジェクト117を出現させるには、オペレータ94は、剣3の刀身部15の縁を撮像ユニット5に向けて、一定以上の速度で振る必要がある。つまり、オペレータ94が、このよう  
5 5に剣3を振ると、剣3の半円柱状部材21の反射シート23が、撮像ユニット5により撮像され、その処理結果を基に剣軌跡オブジェクト117のトリガが発生する。

さて、オペレータ94が剣3を振ったときに出現する剣軌跡オブジェクト117の一部が、敵オブジェクト115を含む所定範囲内に存在する場合は、図15に示すように、エフェクト119が与えられた敵オブジェクト121が表示される。こうすることで、オペレータ94は、剣軌跡オブジェクト11  
10 7が敵オブジェクト115にヒットしたことを判断できる。敵オブジェクト115への連続したヒット回数が所定値を超えると、強さ情報が更新され、強さがアップする。強さ情報には、例えば、生命力を表現するライフ情報、及び、使用できる特殊技の数を表現するポイント情報、等が含まれる。このような強さ情報は、例えば、対戦モードを実行するときのために、メモ리카ートリッジ13に格納される。

また、オペレータ94が、剣3の刀身部15の側面を撮像ユニット5に向けると、図16に示すよう  
15 15に、盾オブジェクト123が出現する。つまり、剣3の刀身部15の側面を撮像ユニット5に向けると、刀身部15の側面に取り付けられた反射シート17が撮像ユニット5により撮影され、その処理結果を基に盾オブジェクト123のトリガが発生する。

この盾オブジェクト123は、剣3の刀身部15の側面を撮像ユニット5に向けて動かすと、その動きに沿って画面上を移動する。従って、剣3を操作して、盾オブジェクト123を動かすことで、敵オブジェクト125の攻撃（図16の例では、炎オブジェクト127）を防御できる。つまり、オペレー  
20 20タ94が、剣3を動かして、盾オブジェクト123を移動させ、敵オブジェクト125からの炎オブジェクト127にタイミングよく盾オブジェクト123を重ねることができれば、炎オブジェクト127は消滅し、敵オブジェクト125の攻撃を防御できる。

さて、ストーリーモードでは、図17に示すような、説明オブジェクト129が、スクリーン91に  
25 25表示される場合がある。この場合、オペレータ94は、説明オブジェクト129の指示に従って、剣3を操作して、ゲームを進める。図17の例では、オペレータ94が剣3を振ると、その時表示されていた説明オブジェクト129が消滅して、次の説明オブジェクトがスクリーン91に表示される。つまり、オペレータ94が、剣3の刀身部15の縁を撮像ユニット5に向けて振ると、剣3の半円柱状部材21

の反射シート23が、撮像ユニット5により撮影され、その処理結果を基に、説明オブジェクト129を次の説明オブジェクトに進めるためのトリガが発生する。

また、ストーリーモードでは、図18(a)に示すような説明オブジェクト132が表示される場合もある。この場合、オペレータ94が、剣3の刃先部を撮像ユニット5に向けると、図18(b)に示すように、オペレータ94が実空間を前進したような画面が表示される。つまり、オペレータ94が、剣3の刃先を撮像ユニット5に向けると、静止した剣3の半円柱状部材21の反射シート23が、撮像ユニット5により撮影され、その処理結果を基に、画面(背景画像)を次に進めるためのトリガが発生する。

さて次に、対戦モードについて説明する。対戦モードでは、情報処理装置1が、2人のオペレータ94のそれぞれのメモ리카ートリッジ13に格納された強さ情報を読み取り、その強さ情報を前提にして、対戦ゲームが実行される。メモ리카ートリッジ13に格納されている強さ情報は、2人のオペレータ94のそれぞれが、ストーリーモードによるゲームで取得した強さ情報である。情報処理装置1は、2人のオペレータ94のそれぞれの強さ情報を読み取ると、次のようなゲーム画面を表示する。

図19は、図12の選択画面で「対戦モード」の項目オブジェクト109が選択されたときのゲーム画面の例示図である。図19に示すように、対戦モードのゲーム画面には、生命力を表現するライフ情報131a、131b、使用できる特殊技の数を表現するポイント情報141a、141b、対戦オブジェクト133a、133b、及び、コマンド選択部135a、135b、が表示される。コマンド選択部135a、135bには、選択枠137a、137b及びコマンドオブジェクト139a、139bが表示される。

ライフ情報131a、131bは、それぞれ、各オペレータ94のメモ리카ートリッジ13から取得したライフ情報である。図19の例では、残りライフを棒グラフで表現している。ポイント情報141a、141bは、それぞれ、各オペレータ94のメモ리카ートリッジ13から取得したポイント情報である。

2人のオペレータ94のいずれかが、剣3を振ると、コマンド選択部135a、135bのコマンドオブジェクト139a、139bが、左方向へ回転する。そして、一方のオペレータ94が、自分の剣3を振って、コマンド選択部135aで回転しているコマンドオブジェクト139aを静止させる。同様に、他方のオペレータ94が、自分の剣3を振って、コマンド選択部135bで回転しているコマンドオブジェクト139bを静止させる。

そして、選択枠137a, 137b内で静止したコマンドオブジェクト139a, 139bに従って対戦処理が実行される。図19の例では、対戦オブジェクト133aは、「無防備」の状態となり、対戦オブジェクト133bから、「攻撃C」を受ける。すると、対戦オブジェクト133aのライフ情報131aが減少する。このように、オペレータ94のそれぞれが静止させたコマンドオブジェクト139a, 139bに従って、対戦が行われる。

攻撃のコマンド139a, 139bの強さは、A、B、Cの順となっている。同様に、防御のコマンド139a, 139bの強さも、A、B、Cの順となっている。

強さに差がある攻撃のコマンドが選択された場合、弱いコマンドを選択したほうがダメージを受け、強さの差に応じてライフ情報が減少する。また、強さが同じ攻撃のコマンドが選択された場合、鎧迫り合いとなる。このときは、所定時間内に、より多く剣3を振ったオペレータの対戦オブジェクトが、振りの少ないオペレータの対戦オブジェクトにダメージを与えることができ、ライフを減少させることができる。

強い攻撃のコマンド及び弱い防御のコマンドが選択された場合、弱い防御のコマンドを選択したほうがダメージを受け、強さの差に応じてライフ情報が減少する。弱い攻撃のコマンド及び強い防御のコマンドが選択された場合、防御側はダメージを受けない。強さが同じ攻撃のコマンド及び防御のコマンドが選択された場合は、両者ダメージを受けない。

ポイント情報141a, 141bは、特殊技を使用したら減少する。特殊技は、特殊技のコマンドオブジェクト139a, 139bを静止させたときに実行される。

さて、次に、情報処理装置1によるゲーム処理の詳細を説明する。

図20は、図7のROM65に格納されたプログラム及びデータを示す概念図である。図20に示すように、ROM65には、制御プログラム102、画像データ103、及び、音声データ105、が格納される。これらのプログラム及びデータの内容は以下の説明の中で明らかになる。

図8のCPU201は、イメージセンサ43が出力したアナログのピクセルデータを変換したデジタルのピクセルデータを取得して、配列P[X][Y]に代入する。なお、上記のように、イメージセンサ43の水平方向（横方向、行方向）をX軸、垂直方向（縦方向、列方向）をY軸としている。

そして、CPU201は、赤外発光ダイオード7の点灯時のピクセルデータP[X][Y]と、消灯時のピクセルデータP[X][Y]と、の差分を算出して、差分データを配列Diff[X][Y]に代入する。ここで、図面を用いて、差分を求める効果を説明する。ここで、ピクセルデータは輝度を表す。よって、差分データも輝度を表す。

図 2 1 (a) は、一般的なイメージセンサにより撮影された、特別な処理を施さない画像の例示図、  
図 2 1 (b) は、図 2 1 (a) の画像信号を或る閾値でレベル弁別したときの画像信号の例示図、図 2  
1 (c) は、赤外フィルタ 9 を介したイメージセンサ 4 3 の点灯時の画像信号を或る閾値でレベル弁別  
したときの画像信号の例示図、図 2 1 (d) は、赤外フィルタ 9 を介したイメージセンサ 4 3 の消灯時  
5 の画像信号を或る閾値でレベル弁別したときの画像信号の例示図、図 2 1 (e) は、点灯時の画像信号  
と消灯時の画像信号との差分信号の例示図、である。

上記のように、剣 3 に赤外光を照射し、赤外フィルタ 9 を介してイメージセンサ 4 3 に入射した反射  
赤外光による画像を撮影している。一般的な室内環境で一般的な光源を用いて剣 3 をストロボスコープ  
撮影した場合には、一般的なイメージセンサ (図 6 のイメージセンサ 4 3 に相当する。) には、図 2 1  
10 (a) に示すように、剣 3 による画像以外に、蛍光灯光源、白熱灯光源、太陽光 (窓) のような光源だ  
けでなく、室内のすべてのものの画像がすべて写り込む。したがって、この図 2 1 (a) の画像を処理  
して剣 3 の画像のみを抽出するのは、かなり高速のコンピュータまたはプロセサが必要である。しかし  
ながら、安価が条件の装置ではそのような高性能コンピュータを使えない。そこで種々の処理を行って  
負担を軽減することが考えられる。

15 なお、図 2 1 (a) の画像は、本来ならば、白黒の階調により表される画像であるが、その図示を省  
略している。また、図 2 1 (a) ~ 図 2 1 (e) は、剣 3 の刀身部 1 5 の縁をイメージセンサに向けた  
場合の画像であるため、反射シート 1 7 ではなく、反射シート 2 3 が撮影されている。なお、2 つの反  
射シート 2 3 の距離が近いので、通常、2 つの反射シート 2 3 は、1 点として撮影される。

さて、図 2 1 (b) は、図 2 1 (a) の画像信号を或る閾値でレベル弁別したときの画像信号である。  
20 このようなレベル弁別処理は専用のハードウェア回路でも、ソフトウェア的にでも、実行することがで  
きるが、いずれの方法によっても、一定以下の光量のピクセルデータをカットするレベル弁別を実行す  
ると、剣 3 や光源以外の低輝度画像を除去することができる。この図 2 1 (b) の画像では剣 3 および  
室内の光源以外の画像の処理を省略でき、したがって、コンピュータの負担を軽減できるが、それでも、  
光源画像を含む高輝度画像が依然として写り込んでいるので、剣 3 と他の光源を分別することは難しい。

25 そこで、図 6 に示したように赤外フィルタ 9 を利用して、イメージセンサ 4 3 に赤外光による画像以  
外の画像が写らないようにした。それによって、図 2 1 (c) に示すように、赤外光を殆ど含まない蛍  
光灯光源の画像は除去できる。しかしながら、それでもなお太陽光や白熱灯が画像信号中に含まれてし  
まう。したがって、更なる負担軽減のために、赤外ストロボスコープの点灯時のピクセルデータと消灯  
時のピクセルデータとの差分を計算することとした。

そのため、図21(c)の点灯時の画像信号のピクセルデータと、図21(d)の消灯時の画像信号のピクセルデータとの差分を計算した。すると、図21(e)に示すように、その差分だけの画像が取得できる。この差分データによる画像は、図21(a)と対比すれば明らかなように、剣3によって得られる画像のみを含むことになる。したがって、処理の軽減を図りながらも、剣3の状態情報を取得できる。状態情報とは、例えば、速さ情報、移動方向情報、移動距離情報、速度ベクトル情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、又は、それらの2以上の組み合わせ、等である。

以上のような理由で、CPU201は、赤外発光ダイオード7の点灯時のピクセルデータと、消灯時のピクセルデータと、の差分を算出して、差分データを得る。

- 10 CPU201は算出した差分データ $Dif[X][Y]$ を基に、剣3の反射面(反射シート17, 23)を検出する。具体的には次の通りである。

イメージセンサ43は、上記のように、例えば、32ピクセル×32ピクセルからなる。CPU201は、X方向に32ピクセル分の差分データをスキャンし、そして、Y座標をインクリメントし、X方向に32ピクセル分の差分データをスキャンし、そして、Y座標をインクリメントし、というように、  
15 Y座標をインクリメントしながら、X方向に32ピクセル分の差分データをスキャンして、所定の閾値 $Th$ より大きい差分データを持つピクセル数をカウントしていく。所定の閾値 $Th$ より大きい差分データを持つピクセルが存在する場合は、反射シート17又は反射シート23が検出されたと判断される。

- そして、CPU201は、所定の閾値 $Th$ を超えた差分データから最大値を求める。最大の差分データを持つピクセルを剣3の注目点とする。従って、注目点のX座標及びY座標は、最大の差分データを持つピクセルのX座標及びY座標である。さらに、CPU201は、イメージセンサ43上(イメージ  
20 センサ43によるイメージ上)のX座標及びY座標を、スクリーン91上(表示画面上)の $x$ 座標及び $y$ 座標に変換して、 $x$ 座標及び $y$ 座標をそれぞれ配列 $Px[M]$ 及び $Py[M]$ に代入する。スクリーン91には、グラフィックプロセッサ202が生成した横256画素×縦224画素の画像が表示される。従って、スクリーン91上の位置 $(x, y)$ は、スクリーン91の中心を原点 $(0, 0)$ として、  
25 画素の位置で示される。なお、「 $M$ 」は整数であり、 $M$ 回目に撮影されたことを意味する。以上のようにして、CPU201は、剣3の注目点を抽出する。

CPU201は、抽出した前回と今回の注目点の座標に基づいて、剣3がスイングされたか否かを判断する。詳細は次の通りである。

CPU201は、現在の注目点 (M) の座標 ( $P_x [M]$  ,  $P_y [M]$  ) と、前回の注目点 (M-1) の座標 ( $P_x [M-1]$  ,  $P_y [M-1]$  ) と、を用いて、次式により、剣3の注目点 (M) の速度ベクトル ( $V_x [M]$  ,  $V_y [M]$  ) を求める。

$$V_x [M] = P_x [M] - P_x [M-1] \quad \cdots (1)$$

5  $V_y [M] = P_y [M] - P_y [M-1] \quad \cdots (2)$

そして、CPU201は、次式により、剣3の注目点 (M) の速さ  $V [M]$  を求める。

$$V [M] = \sqrt{(V_x [M]^2 + V_y [M]^2)} \quad \cdots (3)$$

CPU201は、注目点 (M) の速さ  $V [M]$  と、所定の閾値  $ThV$  と、を比較して、速さ  $V [M]$  が大きい場合は、剣3が振られたと判断して、スイングフラグをオンにする。

10 CPU201は、剣3のスイング方向を検出する。詳細は次の通りである。

図22は、図8のCPU201が剣3のスイング方向を検出する際の説明図である。図22に示すように、スクリーン91の中心を原点とし、256画素×256画素の仮想平面を想定する。仮想平面上の座標は、スクリーン91上の座標と一致する。この仮想平面の範囲外に、仮想の注目点 (0) を設定し、この注目点の座標を、( $P_x [0]$  ,  $P_y [0]$  ) とする。

15 注目点 (1) の速さ  $V [1]$  が所定の閾値  $ThV$  を超えたとする。そして、注目点 (2) の速さ  $V [2]$  及び注目点 (3) の速さ  $V [3]$  も引き続き、所定の閾値  $ThV$  をこえており、注目点 [4] の速さ  $V [4]$  が所定の閾値  $ThV$  以下になったとする。

CPU201は、所定の閾値  $ThV$  を最初に超えた注目点 (1) の座標 ( $P_x [1]$  ,  $P_y [1]$  ) と、最初に所定の閾値  $ThV$  以下になった注目点 (4) の座標 ( $P_x [4]$  ,  $P_y [4]$  ) と、に基づいて、剣3のスイング方向を検出する。具体的には、次の通りである。なお、速さが所定の閾値  $ThV$  を最初に超えた注目点 (S) のx座標及びy座標をそれぞれ  $P_x [S]$  及び  $P_y [S]$  とし、速さが最初に所定の閾値  $ThV$  以下になった注目点 (E) のx座標及びy座標をそれぞれ  $P_x [E]$  及び  $P_y [E]$  とする。

CPU201は、次式により、これら2点間の距離を求める。

25  $L_x = P_x [E] - P_x [S] \quad \cdots (4)$

$$L_y = P_y [E] - P_y [S] \quad \cdots (5)$$

そして、距離  $L_x$  ,  $L_y$  を、所定の閾値  $ThV$  を超えた注目点の数「n」で除算する。図22の例では、 $n=3$  である。

$$L_x A = L_x / n \quad \cdots (6)$$



$$L_y A = L_y / n \quad \cdots (7)$$

なお、CPU 201は、所定の閾値ThVを最初に超えた注目点（S）から、イメージセンサ43の撮影範囲内の注目点（図22の例では注目点（4））まで、全てが所定の閾値ThVを超えており、所定の閾値ThV以下にならなかった場合は、イメージセンサ43の撮影範囲外になる直前に抽出された

5 注目点（図22の例では注目点（4））を注目点（E）として、この注目点（E）と所定の閾値ThVを最初に超えた注目点（S）と、を基に、式（4）～式（7）の計算を実行する。なお、この場合、 $n = n - 1$ となる。

次に、CPU 201は、x方向のスイング長の平均値 $L_x A$ の絶対値と所定値 $x_r$ との間で大小を判断する。また、CPU 201は、y方向のスイング長の平均値 $L_y A$ の絶対値と所定値 $y_r$ との間で大小を判断する。その判断の結果、CPU 201は、平均値 $L_x A$ の絶対値が所定値 $x_r$ より大きく、平均値 $L_y A$ の絶対値が所定値 $y_r$ より小さい場合は、剣3が横方向（水平方向）に振られたと判断して、角度フラグを対応する値にセットする。

また、判断の結果、CPU 201は、平均値 $L_x A$ の絶対値が所定値 $x_r$ より小さく、平均値 $L_y A$ の絶対値が所定値 $y_r$ より大きい場合は、剣3が縦方向（垂直方向）に振られたと判断して、角度フラグを対応する値にセットする。また、判断の結果、CPU 201は、平均値 $L_x A$ の絶対値が所定値 $x_r$ より大きく、平均値 $L_y A$ の絶対値が所定値 $y_r$ より大きい場合は、剣3が斜め方向に振られたと判断して、角度フラグを対応する値にセットする。

さらに、CPU 201は、平均値 $L_x A$ の符号を判断して、x方向フラグを対応する値にセットする。また、CPU 201は、平均値 $L_y A$ の符号を判断して、y方向フラグを対応する値にセットする。なお、x方向フラグ及びy方向フラグを包括して表現するときは、単に方向フラグと呼ぶ。

CPU 201は、角度フラグ、x方向フラグ、及び、y方向フラグ、にセットされている値に従って、剣3のスイング情報を決定する。剣3のスイング情報は、剣3のスイング方向を表す情報である。このスイング情報により、剣軌跡プロジェクト117の種類が決定される。この点の詳細を説明する。

図23（a）は、角度フラグの値と角度との関係図、図23（b）は、方向フラグの値と方向を表す符号との関係図、図23（c）は、角度フラグ及び方向フラグと、スイング情報と、の関係図、である。上記のように、CPU 201は、平均値 $L_x A$ 及び平均値 $L_y A$ の絶対値と、所定値 $x_r$ 及び所定値 $y_r$ と、の間で大小を判断して、図23（a）に示すように、角度フラグをセットする。

また、上記のように、CPU 201は、平均値 $L_x A$ 及び平均値 $L_y A$ の符号を判断して、図23（b）に示すように、x方向フラグ及びy方向フラグをセットする。

さらに、図23(c)に示すように、CPU201は、角度フラグ、x方向フラグ、及び、y方向フラグ、にセットされている値から、剣3のスイング情報を決定する。

図24は、図23(c)のスイング情報と、剣3の操作方向と、の関係図である。図23及び図24に示すように、スイング情報A0は、剣3が、横方向に、かつ、x軸の正方向(右方向)に操作されたことを意味する。スイング情報A1は、剣3が、横方向に、かつ、x軸の負方向(左方向)に操作されたことを意味する。スイング情報A2は、剣3が、縦方向に、かつ、y軸の正方向(上方向)に操作されたことを意味する。スイング情報A3は、剣3が、縦方向に、かつ、y軸の負方向(下方向)に操作されたことを意味する。スイング情報A4は、剣3が、右上斜め方向に操作されたことを意味する。スイング情報A5は、剣3が、右下斜め方向に操作されたことを意味する。スイング情報A6は、剣3が、左上斜め方向に操作されたことを意味する。スイング情報A7は、剣3が、左下斜め方向に操作されたことを意味する。

CPU201は、上記のようにして取得したスイング情報A0～A7に関連付けられたアニメーションテーブル格納位置情報を登録する(剣軌跡登録：トリガ発生を意味)。アニメーションテーブル格納位置情報は、アニメーションテーブルの格納位置を示す情報である。また、この場合のアニメーションテーブルには、剣軌跡オブジェクト117をアニメーションするための様々な情報が含まれている。

なお、注目点の速さ情報が、所定の閾値ThVを超えてから、所定の閾値ThV以下になるまでの、注目点の数が3以上の場合に、上記のアニメーションテーブル格納位置情報の登録が行われ、3より小さい場合は、登録は行われない。つまり、注目点の数が、2点以下の場合には、上記登録は行われない。また、所定の閾値ThVを最初に超えた注目点から、イメージセンサ43の撮影範囲内の注目点まで、全てが所定の閾値ThVを超えており、所定の閾値ThV以下にならなかった場合も同様に、注目点の数が3以上の場合に、上記のアニメーションテーブル格納位置情報の登録が行われ、3より小さい場合は、登録は行われない。

図25は、スイング情報A0～A7とアニメーションテーブル格納位置情報との関係図である。図25では、例えば、スイング情報A0、A1とアニメーションテーブル格納位置情報address0とが関連付けられている。ここで、アニメーションテーブル格納位置情報は、アニメーションテーブルが格納されている領域の先頭アドレス情報である。

図26は、剣軌跡オブジェクト117をアニメーションするためのアニメーションテーブルの例示図である。図26に示すように、アニメーションテーブルは、画像格納位置情報、コマ指定情報、持続フレーム数情報、及び、サイズ情報、からなる。画像格納位置情報は、画像データの格納位置を示す情報

である。この画像データは、アニメーションを行うためのものであるため、コマごとのオブジェクト画像データからなる。なお、画像格納位置情報は、最初のコマのオブジェクト画像データが格納されている領域の先頭アドレス情報である。コマ指定情報は、何コマ目のオブジェクト画像データであるかを示す情報である。持続フレーム数情報は、コマ指定情報が指定するコマのオブジェクト画像データを、何フレーム継続して表示するかを示す情報である。サイズ情報は、オブジェクト画像データのサイズを示す情報である。

ここで、図26のアニメーションテーブルは、剣軌跡オブジェクト117のアニメーションを行うためのものである。従って、例えば、スイング情報A0, A1は、剣3が横方向に振られたことを示す情報であるため、アニメーションテーブル格納位置情報address0が示すアニメーションテーブルの画像格納位置情報a0は、横方向の剣軌跡を表す剣軌跡オブジェクト117の格納位置を示す。

図27(a)～図27(m)は、剣軌跡オブジェクト117のアニメーションを行うためのオブジェクト画像データの例示図である。図27(a)～図27(m)のそれぞれが、コマに相当する。図27(a)～図27(m)に示すように、最初は幅wの狭い帯状の画像(剣軌跡オブジェクト117)が、コマ(時間t)が進むに従って、幅wが広くなり、さらに、コマが進むに従って、幅wが小さくなる。この例は、スイング情報A0, A1に対応する図26の画像格納位置情報a0が示す位置に格納された画像データの1例である。なお、画像格納位置情報a0は、図27(a)のオブジェクト画像データの先頭アドレスを示す。

ここで、スプライト及び背景について簡単に説明する。剣軌跡オブジェクト117や盾オブジェクト123等のオブジェクトは、単数又は複数のスプライトから構成される。プライトは、スクリーン91のいずれの位置にでも配置可能な1つの矩形の画素集合(例えば、16画素×16画素)からなる。一方、背景は、矩形の画素集合(例えば、16画素×16画素)の二次元配列からなり、スクリーン91全体を覆う大きさを持つ(例えば、横256画素×縦256画素)。スプライトや背景を構成する矩形の画素集合を、キャラクタと呼ぶ。

図27(a)のオブジェクト画像データを構成する各スプライトの格納位置情報(先頭アドレス)は、剣軌跡オブジェクト117の格納位置情報a0とスプライトのサイズとから算出される。また、図27(b)～図27(m)の各オブジェクト画像データの格納位置情報(先頭アドレス)は、画像格納位置情報a0並びにアニメーションテーブルのコマ指定情報及びサイズ情報から算出される。オブジェクト画像データの各々を構成する各スプライトの格納位置情報(先頭アドレス)は、各オブジェクト画像デ

ータの格納位置情報とスプライトのサイズとから算出される。ただし、オブジェクト画像データや各スプライトの格納位置情報を計算で求めるのではなく、アニメーションテーブルに予め持っていてよい。

なお、図27(a)～図27(m)において、黒く塗りつぶした部分は、透明であることを意味する。また、ハッチングの種類の違いは、色の違いを示している。さらに、この例では、1コマを1フレームだけ表示するため、13コマを13フレームで表示する。なお、フレームの更新は、例えば、1/60秒ごとに行われる。以上のように、コマ(時間t)の進行とともに、剣軌跡オブジェクト117の幅wを、小→大→小、と変化させることで、剣3が振られたことに応じて、鋭い閃光が走ったような剣軌跡の表現が可能となる。

図28(a)～図28(m)は、剣軌跡オブジェクト117のアニメーションを行うためのオブジェクト画像データの他の例示図である。図28(a)～図28(m)に示すように、最初は幅wの広い帯状の画像(剣軌跡オブジェクト117)が、コマ(時間t)が進むに従って、幅wが狭くなる。また、最初は短い剣軌跡オブジェクト117であるが、コマが進むに従って、長くなり、そして、一定長さになっている。なお、この例は、スイング情報A1に対応する剣軌跡オブジェクト117のアニメーションのためのオブジェクト画像データの1例である。従って、剣軌跡画像は、剣3が移動する方向(図24参照)に対応して右側から出現している。ここで、スイング情報がA0の場合は、図28(a)～図28(m)のオブジェクト画像データの方向が逆になる。つまり、図28(a)～図28(d)においては、剣軌跡画像が左側から出現する。同様に、他のスイング情報A2～A7に対応するオブジェクト画像データにおいても、剣3が移動する方向(図24参照)に対応した方向から剣軌跡画像が出現する。

図29(a)～図29(m)は、剣軌跡オブジェクト117のアニメーションを行うためのオブジェクト画像データの他の例示図である。図29(f)～図29(m)に示すように、幅wの画像(白で表記)に、残像(ハッチングで表記)を付加することもできる。なお、この例は、スイング情報A1に対応する剣軌跡オブジェクト117のアニメーションのためのオブジェクト画像データの1例である。従って、剣軌跡画像は、剣3が移動する方向(図24参照)に対応して右側から出現している。ここで、スイング情報がA0の場合は、図29(a)～図29(m)のオブジェクト画像データの方向が逆になる。つまり、図29(a)～図29(d)においては、剣軌跡画像が左側から出現する。同様に、他のスイング情報A2～A7に対応するオブジェクト画像データにおいても、剣3が移動する方向(図24参照)に対応した方向から剣軌跡画像が出現する。

なお、図27～図29において、剣軌跡画像を白色で表している部分が存在するが、実際には、所望の色(白色含む。)が付される。

さて、CPU201は、剣軌跡オブジェクト117のスクリーン91上の座標を算出する。まず、スイング情報が「A0」又は「A1」の例を挙げる。従って、CPU201は、速さが所定の閾値ThVを最初に超えた注目点(S)のy座標(Py[S])と、速さが最初に所定の閾値ThV以下になった注目点(E)のy座標(Py[E])と、を用いて、剣軌跡オブジェクト117の中心のy座標(y

5 t)を決定する。つまり、次式の通りである。

$$y_t = (P_y[S] + P_y[E]) / 2 \quad \dots (8)$$

一方、剣軌跡オブジェクト117の中心のx座標(xt)は、次式のようにになる。

$$x_t = 0 \quad \dots (9)$$

こうすれば、剣軌跡オブジェクト117が出現する縦方向(垂直方向)位置は、オペレータ94による剣3の操作に応じたものとなる。一方、この例では、スイング情報が「A0」又は「A1」であるため、すなわち、剣3が横方向に振られているため、剣軌跡オブジェクト117の中心のx座標(xt)は、画面の中心のx座標、即ち、「0」とすることが適切である。

次に、スイング情報が「A2」あるいは「A3」の場合、つまり、剣3が縦方向に振られた場合、を説明する。この場合、速さが所定の閾値ThVを最初に超えた注目点(S)のx座標をPx[S]とし、速さが最初に所定の閾値ThV以下になった注目点(E)のx座標をPx[E]とする。そうすると、剣軌跡オブジェクト117の中心の座標(xt, yt)は、次式のようにになる。

$$x_t = (P_x[S] + P_x[E]) / 2 \quad \dots (10)$$

$$y_t = 0 \quad \dots (11)$$

こうすれば、剣軌跡オブジェクト117が出現する横方向(水平方向)の位置は、オペレータ94による剣3の操作に応じたものとなる。一方、この例では、スイング情報が「A2」又は「A3」であるため、すなわち、剣3が縦方向に振られているため、剣軌跡オブジェクト117の中心のy座標(yt)は、画面の中心のy座標、即ち、「0」とすることが適切である。

次に、スイング情報が「A4」あるいは「A7」の場合、つまり、剣3が斜め右上方向あるいは斜め左下方向に振られた場合、を説明する。この場合、CPU201は、剣軌跡オブジェクト117の中心座標算出のため、次式により、仮の座標(xs, ys)を求める。

$$x_s = (P_x[S] + P_x[E]) / 2 \quad \dots (12)$$

$$y_s = (P_y[S] + P_y[E]) / 2 \quad \dots (13)$$

そして、CPU201は、座標(xs, ys)を通る直線と、スクリーン91の右下がりの対角線との交点座標(xI, yI)を求める。この場合の、座標(xs, ys)を通る直線は、スクリーン91

の右上がりの対角線と平行な直線である。なお、厳密な交点座標  $(x_I, y_I)$  を求める必要は必ずしもない。こうして求めた交点座標  $(x_I, y_I)$  を、剣軌跡オブジェクト117の中心座標  $(x_t, y_t)$  とする。

5     スイング情報が「A5」あるいは「A6」の場合、つまり、剣3が斜め右下方向あるいは斜め左上方向に振られた場合、CPU201は、仮の座標  $(x_s, y_s)$  を通る直線と、スクリーン91の右上がりの対角線と、の交点座標  $(x_I, y_I)$  を求める。この場合の、座標  $(x_s, y_s)$  を通る直線は、画面の右下がりの対角線と平行な直線である。なお、厳密な交点座標  $(x_I, y_I)$  を求める必要は必ずしもない。こうして求めた交点座標  $(x_I, y_I)$  を、剣軌跡オブジェクト117の中心座標  $(x_t, y_t)$  とする。

10     なお、CPU201は、所定の閾値  $ThV$  を最初に超えた注目点 (S) から、イメージセンサ43の撮影範囲内の注目点 (図22の例では注目点 (4)) まで、全てが所定の閾値  $ThV$  を超えており、所定の閾値  $ThV$  以下にならなかった場合は、イメージセンサ43の撮影範囲外になる直前に抽出された注目点 (図22の例では注目点 (4)) を注目点 (E) として、この注目点 (E) と所定の閾値  $ThV$  を最初に超えた注目点 (S) と、を基に、式 (8) ~ 式 (13) の計算を実行する。

15     次に、剣軌跡オブジェクト117が、敵オブジェクト115にヒットしたかどうかの判定手法を説明する。

図30は、図8のCPU201によるヒット判定の説明図である。図30に示すように、図22と同じ仮想平面を想定する。また、スイング情報が「A0」又「A1」の剣軌跡オブジェクト117の長手方向の中心線327を想定する。さらに、この中心線327に中心座標が存在する5つの仮想矩形329~337を想定する。ここで、各仮想矩形329~337の頂点座標を包括して、座標  $(x_{pq}, y_{pq})$  と表記する。ここで、「p」は、各仮想矩形329~337を示しており、図30の例では、 $p = 1 \sim 5$ 、である。また、「q」は、仮想矩形329~337の各々において、各頂点を示しており、図30の例では、 $q = 1 \sim 4$ 、である。

25     一方、m番目 (mは自然数) の敵オブジェクト115の中心座標を中心として、ヒットレンジ325を想定する。また、m番目のヒットレンジ325の各頂点の座標を、 $(x_{m1}, y_{m1})$ 、 $(x_{m1}, y_{m2})$ 、 $(x_{m2}, y_{m2})$ 、 $(x_{m2}, y_{m1})$ 、とする。

CPU201は、 $x_{m1} < x_{pq} < x_{m2}$ 、を満たし、かつ、 $y_{m1} < y_{pq} < y_{m2}$ 、を満たすかどうかを、全ての仮想矩形329~337の全ての頂点座標  $(x_{pq}, y_{pq})$  について判断する。そして、CPU201は、このような条件を満たした頂点座標  $(x_{pq}, y_{pq})$  が存在する場合は、m

番目の敵オブジェクト115に剣軌跡オブジェクト117がヒットしたと判定する。つまり、仮想矩形329～337のうちのいずれかが、ヒットレンジ325と重なった場合は、ヒットしたと判定される。

上記のような判定が、表示されている全ての敵オブジェクト115に対して行われる。また、スイング情報が、「A2」～「A7」の場合も、「A0」及び「A1」の場合と同様であり、仮想矩形がヒットレンジと重なったかどうかで、ヒット判定がなされる。なお、仮想矩形及びヒットレンジが実際に画像として表示されるわけではなく、あくまでも仮想のものである。

また、CPU201は、ヒットと判定したときは、エフェクト119を表示するためのヒット登録（トリガ発生を意味）を行う。具体的には、CPU201は、ヒットしたときのスイング情報「A0」～「A7」に関連付けられたアニメーションテーブル格納位置情報を登録する。この場合のアニメーションテーブル格納位置情報は、エフェクト119のアニメーションを行うためのアニメーションテーブルの格納位置情報である。エフェクト119には方向があるため、スイング情報「A0」～「A7」の各々に対して、アニメーションテーブル格納位置情報が関連付けられている。図15のエフェクト119は、スイング情報「A0」に関連付けられたアニメーションテーブル格納位置情報が示す位置に格納されたアニメーションテーブルに基づく画像である。なお、エフェクト119のアニメーションテーブルは、剣軌跡オブジェクト117のアニメーションテーブルと同様に、画像格納位置情報、コマ指定情報、持続フレーム数情報、及び、サイズ情報、からなる。

また、CPU201は、ヒットと判定したときは、敵オブジェクト115の座標を基に、出現させるエフェクト119の座標を算出する。なぜなら、エフェクト119は、ヒットした敵オブジェクト115の位置に出現させるからである。

次に、盾オブジェクト123の制御について説明する。CPU201は、所定の閾値Thより大きい差分データを持つピクセル数と、所定の閾値ThAと、を比較する。そして、CPU201は、所定の閾値Thより大きい差分データを持つピクセル数が、所定の閾値ThAより大きかった場合、反射シート17、つまり、剣3の刀身部15の側面、が検出されたと判断する。即ち、所定の閾値Thより大きい差分データを持つピクセル数が、所定の閾値ThAより多いということは、赤外光を反射する面積が広いということであるから、検知された反射シートは、面積が小さい反射シート23ではなく、面積が大きい反射シート17ということになる。

CPU201は、面積の大きい反射シート17を検出したときは、盾オブジェクト123を表示するための盾登録（トリガ発生を意味）を行う。具体的には、CPU201は、盾オブジェクト123のアニメーションのためのアニメーションテーブル格納位置情報を登録する。なお、盾オブジェクト123

のアニメーションテーブルは、剣軌跡オブジェクト117のアニメーションテーブルと同様に、画像格納位置情報、コマ指定情報、持続フレーム数情報、及び、サイズ情報、からなる。

また、CPU201は、面積の大きい反射シート17が最初に検出されたときの注目点の座標を、最初の盾オブジェクト123の座標（ $x_s$ ， $y_s$ ）として設定する。

- 5 さらに、CPU201は、剣3の移動に応じて、盾オブジェクト123を移動させるべく、移動後の盾オブジェクト123の座標を算出する。具体的には、次の通りである。ここで、移動後の剣3の注目点の座標を（ $P_x(M)$ ， $P_y(M)$ ）とする。

そうすると、CPU201は、次式により、まず、 $x$ 方向の移動距離 $l_x$ 及び $y$ 方向の移動距離 $l_y$ を求める。なお、次式において、「 $N$ 」は2以上の整数であり、所定値である。

10  $l_x = (P_x[M] - x_s) / N \cdots (14)$

$$l_y = (P_y[M] - y_s) / N \cdots (15)$$

そして、CPU201は、前回の盾オブジェクト123の座標（ $x_s$ ， $y_s$ ）から、この移動距離 $l_x$ ， $l_y$ だけ移動させた座標を移動後の盾オブジェクト123の座標（ $x_s$ ， $y_s$ ）とする。つまり、CPU201は、次式により、移動後の盾オブジェクト123の座標を計算する。

15  $x_s = l_x + x_s \cdots (16)$

$$y_s = l_y + y_s \cdots (17)$$

次に、説明オブジェクト129の制御について説明する。CPU201は、説明オブジェクト129が表示されているときに、剣3が縦方向に振り下ろされた場合は、説明進行登録（トリガ発生を意味）を行う。具体的には、CPU201は、次の説明オブジェクト129を表示するためのアニメーション  
20 テーブル格納位置情報を登録する。なお、説明オブジェクト129のアニメーションテーブルは、剣軌跡オブジェクト117のアニメーションテーブルと同様に、画像格納位置情報、コマ指定情報、持続フレーム数情報、及び、サイズ情報、からなる。ここで、説明オブジェクト129のように、アニメーションを行わない静止画の場合は、コマは1つであり、また、持続フレーム数情報として、最大の値を入れておき、さらに、ループするようにしている。こうすることで、アニメーションテーブルを用いて、  
25 静止画を表示できる。

次に、前進制御について説明する。CPU201は、スクリーン91に前進を指示する表示がなされている場合に、剣3の注目点が、所定フレーム数の間、スクリーン91の中心座標を中心にした所定範囲に存在すれば、前進登録（トリガ発生を意味）を行う（図18（a）及び図18（b）参照）。



CPU201は、前進登録がされたことを条件として、仮想空間内を前進した距離に基づいて、背景を更新する。例えば、仮想空間内を一定距離進むたびに背景を更新する。具体的には次の通りである。

内部メモリ207には、背景を構成する全キャラクタ数と同じ要素数の配列が用意される。そして、配列には、対応するキャラクタの格納位置情報（先頭アドレス）が代入される。従って、背景を更新する

5 ときは、配列の全要素を更新する。

次に、カーソル101の制御について説明する。CPU201は、選択画面において、剣3の注目点を検出したときに、カーソル登録（トリガ発生を意味）を行う（図12参照）。具体的には、CPU201は、カーソル101のアニメーションのためのアニメーションテーブル格納位置情報を登録する。

10 なお、カーソル101のアニメーションテーブルは、剣軌跡オブジェクト117のアニメーションテーブルと同様に、画像格納位置情報、コマ指定情報、持続フレーム数情報、及び、サイズ情報、からなる。

また、CPU201は、剣3の注目点の座標を、最初のカーソル101の座標として設定する。さらに、CPU201は、剣3の移動に応じて、カーソル101を移動させるべく、移動後のカーソル101の座標を算出する。この算出方法は、移動後の盾オブジェクト123の座標の算出方法と同様であり説明を省略する。

15 次に、項目オブジェクト109の制御について説明する。CPU201は、選択画面において、左移動指示オブジェクト103を中心とした所定範囲R1あるいは右移動指示オブジェクト105を中心とした所定範囲R2、にカーソル101が存在するかどうかを判断する。CPU201は、カーソル101が、所定範囲R1内に存在する場合、各項目オブジェクト109の静止位置のx座標から所定値vを差し引く。同様に、CPU201は、カーソル101が、所定範囲R2内に存在する場合、各項目オブジェクト109の静止位置のx座標に所定値vを加える。以上のようにして、移動後の各項目オブジェクト109のx座標を求める。この場合、y座標は固定である。また、項目オブジェクト109が、画面外に移動した場合は、再び右側から出現するように（ループするように）、x座標が設定される。

20 また、CPU201は、項目オブジェクト109の登録を行う。具体的には、CPU201は、項目オブジェクト109の表示のためのアニメーションテーブル格納位置情報を登録する。なお、項目オブジェクト109のアニメーションテーブルは、剣軌跡オブジェクト117のアニメーションテーブルと同様に、画像格納位置情報、コマ指定情報、持続フレーム数情報、及び、サイズ情報、からなる。ただし、説明オブジェクト129と同様に、項目オブジェクト109のアニメーションは行わない。

次に、スイング補正について説明する。CPU201は、x方向の補正情報Kx及びy方向の補正情報Kyを取得する。そして、CPU201は、注目点の座標（x，y）に、これらの補正情報Kx，K

yを加味して、注目点の座標 ( $P_x [M]$  ,  $P_y [M]$  ) とする。つまり、CPU 201は、 $P_x [M] = x + K_x$ 、 $P_y [M] = y + K_y$ 、とする。以下、補正情報の取得について詳細に説明する。

図31は、図12の選択画面で「スイング補正」の項目オブジェクト109が選択されたときの、スイング補正画面の例示図である。図31に示すように、スクリーン91のスイング補正画面には、円形オブジェクト111及び説明オブジェクト113が表示される。オペレータ94は、説明オブジェクト113の説明に従って、画面中心に位置する円形オブジェクト111をねらって、剣3を縦（垂直方向）又は横（水平方向）に振る。

オペレータ94が、中心と思って剣3を振った場合でも、イメージセンサ43の向き及び位置と剣3を振る位置との関係により、必ずしも、剣軌跡オブジェクト117が、スクリーン91の中心に表示されるとは限らない。つまり、円形オブジェクト111をねらって、剣3を縦に振った場合でも、x方向に所定距離ずれて、剣軌跡オブジェクト117が表示されたり、また、円形オブジェクト111をねらって、剣3を横に振った場合でも、y方向に所定距離ずれて、剣軌跡オブジェクト117が表示されたり、する場合もある。このずれが、補正情報 $K_x$ 、 $K_y$ であり、これを、剣3の注目点の座標に補正してやれば、オペレータ94がねらった位置に剣軌跡オブジェクト117を出現させることができる。

ここで、剣3を一振りしたときに、注目点が複数検出される。そうすると、縦振りのときは、各注目点のx座標の平均値 $x_A$ を用いて、 $K_x = x_c - x_A$ 、とする。また、横振りのときは、各注目点のy座標の平均値 $y_A$ を用いて、 $K_y = y_c - y_A$ 、とする。ここで、座標( $x_c$ ,  $y_c$ )は、スクリーン91の中心座標(0, 0)である。

さて、これまで説明してきた剣軌跡オブジェクト117等の各オブジェクトの座標は、例えば、そのオブジェクトの中心の座標として定義することができる。また、スプライトの座標は、そのスプライトの中心座標として定義することができる。なお、オブジェクトの座標を、例えば、そのオブジェクトを構成する複数のスプライトのうち、左上角のスプライトの中心座標として定義することもできる。

次に、図1の情報処理装置1の全体の処理の流れを、フローチャートを用いて説明する。

図32は、図1の情報処理装置1の全体の処理の流れを示すフローチャートである。図32に示すように、ステップS1にて、CPU 201は、システムの初期設定を実行する。

ステップS2にて、CPU 201は、ゲーム状態をチェックする。ステップS3にて、CPU 201は、ゲームが終了したかどうかを判断する。ゲームが終了していない場合は、CPU 201は、ステップS4に進み、ゲーム終了の場合は処理を終了する。

ステップS 4にて、CPU 201は、現在のステートを判断する。モード選択のステートであれば、ステップS 5に進み、スイング補正モードであれば、ステップS 6に進み、ストーリーモードであれば、ステップS 7に進み、対戦モードであれば、ステップS 8に進む。なお、ステップS 8では、CPU 201は、対戦モードのゲーム処理を実行する（図19参照）。

- 5     ステップS 9では、CPU 201は、ビデオ同期の割込み待ちかどうかを判断する。本実施の形態では、CPU 201は、テレビジョンモニタ90の表示画面を更新するため画像データを、垂直ブランキング期間の開始後にグラフィックプロセッサ202に与える。従って、表示画面を更新するための演算処理が完了したら、ビデオ同期割込みがあるまで処理を進めないようにしている。

- 10     ステップS 9で「YES」であれば、即ち、ビデオ同期の割込み待ちであれば（ビデオ同期信号による割り込みがなければ）、同じステップS 9に戻る。一方、ステップS 9で「NO」であれば、即ち、ビデオ同期の割込み待ちでなければ（ビデオ同期信号による割り込みがあれば）、ステップS 10に進む。

- 15     ステップS 10では、CPU 201は、ステップS 5～S 8による処理結果に基づいて、画像表示処理を実行し、その後ステップS 2に進む。この場合の画像表示処理とは、表示対象の全スプライトの画像情報（各スプライトの格納位置情報及び座標情報）の取得指示および背景表示のための配列の全要素の取得指示をグラフィックプロセッサ202に与えることを言う。グラフィックプロセッサ202は、これらの情報を取得して、必要な処理を施し、各オブジェクトや背景を表す映像信号を生成する。

- 20     図33は、図32のステップS 1の初期設定処理の流れを示すフローチャートである。図33に示すように、ステップS 20にて、CPU 201は、イメージセンサ43の初期設定処理を行う。ステップS 21にて、CPU 201は、各種フラグ及びカウンタを初期化する。

ステップS 22にて、CPU 201は、タイマ回路210を発音のための割込み源としてセットする。この割込み処理により、音声処理が実行されて、テレビジョンモニタ90のスピーカから、効果音や音楽などの音声が発生される。具体的には次の通りである。

- 25     サウンドプロセッサ203は、タイマ割込みに応じたCPU 201の指示により、内部メモリ207から音声データ105の格納位置情報を取得する。

サウンドプロセッサ203は、取得した格納位置情報を基に、ROM65から音声データ105を読み出して、必要な処理を施し、効果音や音楽などの音声信号を生成する。サウンドプロセッサ203は、生成した音声信号を、音声信号出力端子63に与える。これにより、テレビジョンモニタ90のスピー

カから、効果音や音楽などの音声が発生される。なお、音声データ 105 には、波形データ（音源データ）、及び／又は、エンベロープデータ、が含まれる。

例えば、CPU 201 は、剣軌跡登録がされた場合に（これをトリガとして）、タイマ割込みに応じて、効果音データの格納位置情報の取得指示を出す。すると、サウンドプロセッサ 203 は、その格納位置情報を取得して、ROM 65 から効果音データを読み出して、効果音の音声信号を生成する。これにより、剣軌跡オブジェクト 117 の出現と同時に、効果音が発生され、オペレータ 94 は、剣 3 を振った実感をより認識できる。

図 3 4 は、図 3 3 のステップ S 20 のセンサ初期設定処理の流れを示すフローチャートである。図 3 4 に示すように、最初のステップ S 30 では、高速プロセッサ 200 は、設定データとして、コマンド “CONF” を設定する。ただし、このコマンド “CONF” は、イメージセンサ 43 に、高速プロセッサ 200 からコマンドを送信する設定モードに入ることを知らせるためのコマンドである。そして、次のステップ S 31 にて、コマンド送信処理を実行する。

図 3 5 は、図 3 4 のステップ S 31 のコマンド送信処理の流れを示すフローチャートである。図 3 5 に示すように、最初のステップ S 40 では、高速プロセッサ 200 は、設定データ（ステップ S 31 の場合はコマンド “CONF”）をレジスタデータ（I/Oポート）に設定し、次のステップ S 41 でレジスタ設定クロック CLK（I/Oポート）をローレベルに設定する。その後、ステップ S 42 で規定時間待機した後、ステップ S 43 で、レジスタ設定クロック CLK をハイレベルに設定する。そして、さらにステップ S 44 での規定時間の待機の後、ステップ S 45 でレジスタ設定クロック CLK を再びローレベルに設定する。

このようにして、図 3 6 に示すように、規定時間の待機を行いながら、レジスタ設定クロック CLK をローレベル、ハイレベルそしてローレベルとすることによって、コマンド（コマンドまたはコマンド + データ）の送信処理が行われる。

図 3 4 の説明に戻る。ステップ S 32 では、ピクセルモードを設定するとともに、露光時間の設定を行う。この実施の形態の場合、イメージセンサ 43 は先に述べたようにたとえば 32 ピクセル × 32 ピクセルの CMOS イメージセンサであるため、設定アドレス “0” のピクセルモードレジスタに 32 ピクセル × 32 ピクセルであることを示す “0h” を設定する。次のステップ S 33 において、高速プロセッサ 200 は、レジスタ設定処理を実行する。

図 3 7 は、図 3 4 のステップ S 33 のレジスタ設定処理の流れを示すフローチャートである。図 3 7 に示すように、最初のステップ S 50 では、高速プロセッサ 200 は、設定データとして、コマンド

“MOV” + アドレスを設定し、次のステップS 5 1で、図3 5で先に説明したコマンド送信処理を実行して、それを送信する。次にステップS 5 2において、高速プロセッサ2 0 0は、設定データとして、コマンド“LD” + データを設定し、次のステップS 5 3でコマンド送信処理を実行して、それを送信する。そして、ステップS 5 4で、高速プロセッサ2 0 0は、設定データとして、コマンド“SET”  
5 を設定し、次のステップS 5 5でそれを送信する。なお、コマンド“MOV”は制御レジスタのアドレスを送信することを示すコマンドで、コマンド“LD”はデータを送信することを示すコマンドで、コマンド“SET”はデータをそのアドレスに実際に設定させるためのコマンドである。なお、この処理は、設定する制御レジスタが複数ある場合には、繰り返し実行される。

図3 4の説明に戻る。ステップS 3 4では、設定アドレスを“1”（露光時間設定レジスタのローニブルのアドレスを示す）とし、最大露光時間を示す“FFh”のローニブルデータ“Fh”を設定すべきデータとして設定する。そして、ステップS 3 5で図3 7のレジスタ設定処理を実行する。同様に  
10 して、ステップS 3 6において、設定アドレスを“2”（露光時間設定レジスタのハイニブルのアドレスを示す）とし、最大露光時間を示す“FFh”のハイニブルデータ“Fh”を設定すべきデータとして設定し、ステップS 3 7でレジスタ設定処理を実行する。

その後、ステップS 3 8で設定終了を示しかつイメージセンサ4 3にデータの出力を開始させるためのコマンド“RUN”を設定し、ステップS 3 9で送信する。このようにして、図3 3に示すステップS 2 0でのセンサ初期設定処理が実行される。ただし、図3 4～図3 7に示す具体例は、使用されるイ  
15 メージセンサ4 3の仕様に応じて、適宜変更され得るものである。

図3 8は、図3 2のステップS 7のストーリーモードの流れを示すフローチャートである。図3 8に  
20 示すように、ステップS 6 0にて、CPU 2 0 1は、ADC 2 0 8からデジタルのピクセルデータを取得する。このデジタルのピクセルデータは、イメージセンサ4 3からのアナログのピクセルデータが、ADC 2 0 8により、デジタルに変換されたものである。

ステップS 6 1にて、注目領域抽出処理が実行される。具体的には、CPU 2 0 1は、赤外発光ダイ  
25 オード7の発光時のピクセルデータと消灯時のピクセルデータとの差分を算出して、差分データを得る。  
そして、CPU 2 0 1は、その差分データと所定の閾値Thとを比較して、所定の閾値Thを超える差分データを持つピクセル数をカウントする。

ステップS 6 2にて、CPU 2 0 1は、所定の閾値Thを超える差分データから最大値を求め、その  
最大の差分データを持つピクセルの座標を、剣3の注目点とする。

ステップS 6 3にて、CPU 2 0 1は、オペレータ 9 4による剣3のスイング動作を検出して、剣3のスイングに応じた剣軌跡オブジェクト1 1 7を表示するためのトリガを発生する。

ステップS 6 4にて、CPU 2 0 1は、剣軌跡オブジェクト1 1 7が、敵オブジェクト1 1 5にヒットしたかどうかを判定して、ヒットしたときに、エフェクト1 1 9を表示するためのトリガを発生する。

- 5     ステップS 6 5にて、CPU 2 0 1は、剣3の刀身部1 5の側面に取り付けられた反射シート1 7を検知したときに、盾オブジェクト1 2 3を表示するためのトリガを発生する。

ステップS 6 6にて、CPU 2 0 1は、説明オブジェクト1 2 9が表示されている場合において、剣3が縦方向に振り下ろされたときに、次の説明オブジェクト1 2 9を表示するためのトリガを発生する。

- 10    ステップS 6 7にて、CPU 2 0 1は、前進指示が表示されている場合において、所定フレーム数の間、剣3の注目点が所定範囲に存在するときに、前進するような背景のアニメーションを行うべく、背景表示のための配列の各要素の更新処理を行う。

- ステップS 6 8にて、CPU 2 0 1は、「M」が所定値「K」より小さいかどうかを判断する。CPU 2 0 1は、「M」が所定値「K」以上である場合、ステップS 6 9に進み、「M」に「0」を代入して、ステップS 7 0に進む。一方、CPU 2 0 1は、「M」が所定値「K」より小さい場合、ステップ  
15    S 6 8からステップS 7 0に進む。この「M」については、後述の説明の中で明らかになる。

ステップS 7 0では、上記処理結果に基づいて、表示対象の全スプライトの画像情報（各スプライトの格納位置情報や表示位置情報など）を内部メモリ 2 0 7にセットする。

- 図 3 9は、図 3 8のステップS 6 0のピクセルデータ群取得処理の流れを示すフローチャートである。図 3 9に示すように、最初のステップS 8 0で、CPU 2 0 1は、ピクセルデータ配列の要素番号としてXに「-1」、Yに「0」を設定する。本実施の形態におけるピクセルデータ配列は、X=0~31、  
20    Y=0~31の2次元配列であるが、前述のように各行の先頭ピクセルのデータとしてダミーデータが出力されるので、Xの初期値として「-1」が設定される。続くステップS 8 1では、ピクセルデータの取得処理を実行する。

- 図 4 0は、図 3 9のステップS 8 1のピクセルデータ取得処理の流れを示すフローチャートである。図 4 0に示すように、最初のステップS 1 0 0で、CPU 2 0 1は、イメージセンサ 4 3からのフレームステータスフラグ信号F S Fをチェックし、ステップS 1 0 1でそのアップエッジ（ローレベルから  
25    ハイレベルへの）が発生したかどうか判断する。そして、ステップS 1 0 1でフラグ信号F S Fのアップエッジを検出すると、次のステップS 1 0 2において、CPU 2 0 1は、ADC 2 0 8に入力されてきたアナログのピクセルデータのデジタルデータへの変換の開始を指示する。その後、ステップS 1 0

3でイメージセンサ43からのピクセルストローブPDSをチェックし、ステップS104でそのストローブ信号PDSのローレベルからハイレベルへのアップエッジが発生したかどうか判断する。

ステップS104で“YES”が判断されると、CPU201は、ステップS105において、 $X = -1$ かどうか、すなわち先頭ピクセルかどうか判断する。先に述べたように、各行の先頭ピクセルはダミーピクセルとして設定されているので、このステップS105で“YES”が判断されると、次のステップS107でそのときのピクセルデータを取得しないで、要素番号Xをインクリメントする。

ステップS105で“NO”が判断されると、行の第2番目以降のピクセルデータであるので、ステップS106およびS108において、そのときのピクセルデータを取得し、テンポラリレジスタ（図示せず）にそのピクセルデータを格納する。その後、図39のステップS82に進む。

10 図39のステップS82では、テンポラリレジスタに格納されたピクセルデータをピクセルデータ配列P[Y][X]に代入する。

続くステップS83でXをインクリメントする。Xが32に満たない場合、前述のS8.1からS83の処理を繰り返し実行する。Xが32の場合、すなわちピクセルデータの取得が行の終端に到達した場合には、続くステップS85でXに「-1」を設定し、ステップS86でYをインクリメントし、次の

15 行の先頭からピクセルデータの取得処理を繰り返す。

ステップS87でYが32の場合、すなわちピクセルデータの取得がピクセルデータ配列P[Y][X]の終端に到達した場合、図38のステップS61に進む。

図41は、図38のステップS61の注目領域抽出処理の流れを示すフローチャートである。図41に示すように、ステップS120にて、CPU201は、イメージセンサ43からの、赤外発光ダイオード7の点灯時のピクセルデータと、赤外発光ダイオード7の消灯時のピクセルデータと、の差分を算出して、差分データを得る。

20 ステップS121にて、CPU201は、配列Diff[X][Y]に、算出した差分データを代入する。ここで、実施の形態では、32ピクセル×32ピクセルのイメージセンサ43を用いているため、 $X = 0 \sim 31$ 、 $Y = 0 \sim 31$ 、である。

25 ステップS122にて、CPU201は、配列Diff[X][Y]の要素を所定の閾値Thと比較する。

ステップS123にて、CPU201は、配列Diff[X][Y]の要素が所定の閾値Thより大きい場合は、ステップS124に進み、所定の閾値Th以下の場合は、ステップS125に進む。

ステップS124にて、CPU201は、所定の閾値Thを超えた差分データ（配列Diff[X][Y]の要素）の数を計数すべく、カウント値cを1つインクリメントする。

CPU201は、配列Dif[X][Y]の全要素について、所定の閾値Thとの比較が終了するまで、ステップS122からステップS124の処理を繰り返す(ステップS125)。

CPU201は、配列Dif[X][Y]の全要素について、所定の閾値Thとの比較が終了した場合は、ステップS126にて、カウント値cが「0」より大きいかどうかを判断する。

- 5 CPU201は、カウント値cが「0」より大きい場合は、図38のステップS62に進む。カウント値cが「0」より大きいということは、剣3の反射面(反射シート17, 23)が検出されたことを意味する。

- 一方、CPU201は、カウント値cが「0」の場合は、ステップS127に進む。カウント値cが「0」ということは、剣3の反射面(反射シート17, 23)が検出されなかったことを意味する。つまり、剣3が、撮像ユニット5の撮影範囲外に存在することを意味する。従って、ステップS127にて、CPU201は、剣3が撮影範囲外であることを示すレンジアウトフラグをオンにする。

図42は、図38のステップS62の注目点抽出処理の流れを示すフローチャートである。図42に示すように、ステップS140にて、CPU201は、レンジアウトフラグをチェックする。

- CPU201は、レンジアウトフラグがオンであれば、図38のステップS63に進む(ステップS141)。なぜなら、剣3が撮像ユニット5の撮影範囲外ならば、注目点の抽出処理を実行する意味はないからである。一方、CPU201は、レンジアウトフラグがオフであれば、つまり、剣3が検出された場合は、ステップS142に進む(ステップS141)。

ステップS142にて、CPU201は、配列Dif[X][Y]の要素(差分データ)から、最大値を検出する。

- 20 ステップS143にて、CPU201は、「M」を1つインクリメントする。なお、「M」は、図33のステップS21にて「0」に初期化されている。

- ステップS144にて、CPU201は、ステップS142で検出された最大の差分データを持つピクセルの座標(X, Y)を、スクリーン91上の座標(x, y)に変換する。つまり、CPU201は、イメージセンサ43による画像(32ピクセル×32ピクセル)の座標空間から、スクリーン91(横256画素×縦224画素)の座標空間への変換を実行する。

ステップS145にて、CPU201は、配列Px[M]に、変換後のx座標に補正情報Kxを加えたものを代入し、配列Py[M]に、変換後のy座標に補正情報Kyを加えたものを代入する。このようにして、剣3の注目点の座標(Px[M], Py[M])が算出される。



図43は、図38のステップS63のスイング検出処理の流れを示すフローチャートである。図43に示すように、ステップS150にて、CPU201は、レンジアウトフラグをチェックする。

CPU201は、レンジアウトフラグがオンであれば、ステップS160に進み、オフであれば、ステップS152に進む。

- 5     ステップS152にて、CPU201は、式(1)及び式(2)により、剣3の注目点( $P_x[M]$ ,  $P_y[M]$ )の速度ベクトル( $V_x[M]$ ,  $V_y[M]$ )を求める。

ステップS153にて、CPU201は、式(3)により、剣3の注目点( $P_x[M]$ ,  $P_y[M]$ )の速さ $V[M]$ を求める。

- 10    ステップS154にて、CPU201は、剣3の注目点( $P_x[M]$ ,  $P_y[M]$ )の速さ $V[M]$ と所定の閾値 $ThV$ とを比較して、その大小を判断する。CPU201は、注目点の速さ $V[M]$ が、所定の閾値 $ThV$ を超えていれば、ステップS155に進み、所定の閾値 $ThV$ 以下の場合は、ステップS162に進む。

ステップS155にて、CPU201は、スイングフラグをチェックする。

- 15    スイングフラグがオンであれば、CPU201は、ステップS159に進み、オフであれば、ステップS157に進む(ステップS156)。

ステップS157にて、CPU201は、スイングフラグをオンにする。つまり、速さ $V[M]$ が所定の閾値 $ThV$ を超えた場合は、剣3が振られたと判断され、スイングフラグがオンにされる。

ステップS158にて、CPU201は、「S」に、最初に所定の閾値 $ThV$ を超えた注目点の要素番号「M」を代入する。

- 20    ステップS159にて、CPU201は、剣3を1回振ったときに検出される注目点の数を計数すべく、注目点カウンタ $n$ (カウント値 $n$ )を1つインクリメントする。この場合に計数されるのは、速さが所定の閾値を超えた注目点のみである(ステップS154)。ステップS159の後、図38のステップS64に進む。

さて、ステップS162にて、CPU201は、スイングフラグをチェックする。

- 25    スイングフラグがオンであれば、CPU201は、ステップS164に進み、オフであれば、ステップS171に進む(ステップS163)。

スイングフラグがオンであって(ステップS163)、速さが所定の閾値 $ThV$ 以下である(ステップS154)ということは、剣3のスイングが終了したことを意味する。従って、ステップS164にて、CPU201は、スイング終了フラグをオンにする。

ステップS165にて、CPU201は、「E」に、最初に所定の閾値 $ThV$ 以下になった注目点の要素番号「M」を代入する。

ステップS166にて、CPU201は、剣3のスイングに応じた剣軌跡オブジェクト117の種類を決定する。

- 5     ステップS167にて、CPU201は、表示すべき剣軌跡オブジェクト117のスクリーン91上の座標を算出する。

ステップS168にて、CPU201は、ステップS166で決定した剣軌跡オブジェクト117のアニメーションのためのアニメーションテーブル格納位置情報を登録する（剣軌跡登録：トリガに相当）。

- 10    ステップS169にて、CPU201は、注目点カウンタ $n$ （カウント値 $n$ ）をリセットする。

ステップS170にて、CPU201は、スイングフラグをオフにする。

さて、ステップS160にて、CPU201は、注目点カウンタ $n$ （カウント値 $n$ ）を1つデクリメントする。この理由は、後述の図44で説明する。

ステップS161にて、CPU201は、オンとなっているレンジアウトフラグをオフにする。

- 15    そして、ステップS162及びS163を経て、スイングフラグがオンであるということは、注目点の速さが所定の閾値 $ThV$ 以下になる前にレンジアウトしたことを意味する。この場合は、上記したように、レンジアウトする直前の注目点を利用して、剣軌跡オブジェクト117の種類及び座標を決定するため、ステップS164～ステップS170の処理が実行される。

さて、一方、ステップS163にて、スイングフラグがオフと判断された場合は、ステップS171

- 20    において、CPU201は、注目点カウンタ $n$ （カウント値 $n$ ）をリセットする。

図44は、図43のステップS166の剣軌跡種類決定処理の流れを示すフローチャートである。図44に示すように、ステップS180にて、CPU201は、注目点カウンタ $n$ をチェックする。

カウント値 $n$ が「1」より大きい場合は、ステップS182に進み、カウント値 $n$ が「1」以下の場合は、ステップS188に進む（ステップS181）。つまり、カウント値 $n$ が2以上、即ち、速さが所定の閾値 $ThV$ より大きい注目点の数が2以上、の場合にステップS182に進む。さらに、言い換えると、速さが所定の閾値 $ThV$ より大きい注目点の数が2以上の場合は、オペレータ94が意図しないスイング（誤動作）ではなく、オペレータ94が意図してスイングしたと判断されて、ステップS182に進む。

ステップS182にて、CPU201は、式(4)及び式(5)により、スイング長 $L_x$ 、 $L_y$ を求める。

5 ステップS183にて、CPU201は、式(6)及び式(7)により、スイング長 $L_x$ 、 $L_y$ の平均値 $L_{xA}$ 、 $L_{yA}$ を求める。なお、注目点の速さが所定の閾値 $ThV$ 以下になる前にレンジアウトした場合は、上記したように、レンジアウトする直前の注目点を利用して、剣軌跡オブジェクト117の種類及び座標を決定する。このときは、注目点カウンタ $n$ の値は、通常の場合より1つ多いため、図43のステップS160では、注目点カウンタ $n$ をデクリメントしている。

10 ステップS184にて、CPU201は、 $x$ 方向のスイング長 $L_x$ の平均値 $L_{xA}$ の絶対値と、所定値 $x_r$ と、を比較する。また、CPU201は、 $y$ 方向のスイング長 $L_y$ の平均値 $L_{yA}$ の絶対値と、所定値 $y_r$ と、を比較する。

ステップS185にて、CPU201は、ステップS184の結果を基に、角度フラグをセットする(図23(a)参照)。

ステップS186にて、CPU201は、スイング長 $L_x$ 、 $L_y$ の平均値 $L_{xA}$ 、 $L_{yA}$ の符号を判断する。

15 ステップS187にて、CPU201は、ステップS186の結果を基に、方向フラグをセットして(図23(b)参照)、図43のステップS167へ進む。

さて、ステップS188にて、CPU201は、注目点カウンタ $n$ をリセットする。ステップS189にて、CPU201は、スイングフラグ及びスイング終了フラグをオフにする。そして、図38のステップS65に処理が進められる。

20 図45は、図43のステップS167の剣軌跡座標算出処理の流れを示すフローチャートである。図45に示すように、ステップS200にて、CPU201は、角度フラグ及び方向フラグに基づいて、スイング情報を決定する(図23(a)～図23(c)参照)。そして、CPU201は、スイング情報が「A0」又は「A1」の場合は、ステップS201に進み、スイング情報が「A2」又は「A3」の場合は、ステップS202に進み、スイング情報が「A4」～「A7」の場合は、ステップS203  
25 に進む。

ステップS201にて、CPU201は、式(8)及び式(9)により、剣軌跡オブジェクト117の中心座標( $x_t$ ,  $y_t$ )を求める。

また、ステップS202にて、CPU201は、式(10)及び式(11)により、剣軌跡オブジェクト117の中心座標( $x_t$ ,  $y_t$ )を求める。

また、ステップS 2 0 3にて、CPU 2 0 1は、式 ( 1 2 ) 及び式 ( 1 3 ) により、仮の座標 (  $x_s$ ,  $y_s$  ) を求めて、これを通る直線と、画面の対角線と、の交点座標 (  $x_I$ ,  $y_I$  ) を求める。

そして、ステップS 2 0 4にて、CPU 2 0 1は、交点座標 (  $x_I$ ,  $y_I$  ) を、剣軌跡オブジェクト 1 1 7 の中心座標 (  $x_t$ ,  $y_t$  ) とする。

5      なお、ステップS 2 0 1, S 2 0 2, S 2 0 4の後、図4 3のステップS 1 6 8に進む。

図4 6は、図3 8のステップS 6 4のヒット判定処理の流れを示すフローチャートである。図4 6に示すように、ステップS 2 1 0にて、スイング終了フラグがオフであれば、ステップS 2 1 1～S 2 2 1の処理をスキップして、図3 8のステップS 6 5に進む。なぜなら、スイング終了フラグがオフということは、注目点の速さが所定の閾値以下になったわけでもなく、また、注目点がレンジアウトもしていない、ということであるから、剣3のスイングが未だ確定しておらず、剣軌跡オブジェクト1 1 7も表示されていないため、敵オブジェクト1 1 5へのヒット判定を行う必要がないからである。

さて、ステップS 2 1 1とステップS 2 2 0との間で、ステップS 2 1 2～ステップS 2 1 9の処理が繰り返される。ここで、「m」は、敵オブジェクト1 1 5の番号を示しており、「i」は、敵オブジェクト1 1 5の数である。従って、ステップS 2 1 2～ステップS 2 1 9の処理が、敵オブジェクト1 1 5の数だけ繰り返される。つまり、全ての敵オブジェクト1 1 5に対して、ヒット判定が実行される。

また、ステップS 2 1 2とステップS 2 1 9との間で、ステップS 2 1 3～ステップS 2 1 8の処理が繰り返される。ここで、「p」は、仮想矩形の番号を示し、「j」は、仮想矩形の数を示す。図3 0の例では、 $j = 5$ である。従って、ステップS 2 1 3～ステップS 2 1 8の処理が、仮想矩形の数だけ繰り返される。つまり、全ての仮想矩形が、敵オブジェクト1 1 5に重なるかどうか判断される。なお、上記したように、仮想矩形は、剣軌跡オブジェクト1 1 7に仮想的に付加された矩形であり、これが敵オブジェクト1 1 5を含むヒットレンジ3 2 5と重なれば、ヒットとなる。

また、ステップS 2 1 3とステップS 2 1 8との間で、ステップS 2 1 4, S 2 1 5の処理が繰り返される。ここで、「q」は、仮想矩形の頂点の番号を示している。従って、ステップS 2 1 5, S 2 1 6の処理が、仮想矩形の頂点の数だけ繰り返される。つまり、仮想矩形のいずれかの頂点が、敵オブジェクト1 1 5を含むヒットレンジ3 2 5に含まれれば、ヒットとなる。

さて、ステップS 2 1 4にて、CPU 2 0 1は、仮想矩形の頂点のx座標 (  $x_{pq}$  ) が、ヒットレンジ3 2 5のx座標の範囲  $x_{m1} \sim x_{m2}$  に入っているかどうかを判断する。範囲内でない場合は、ステップS 2 1 8に進み、範囲内の場合は、ステップS 2 1 5に進む。

ステップS 2 1 5にて、CPU 2 0 1は、仮想矩形の頂点のy座標 ( $y_{pq}$ ) が、ヒットレンジ3 2 5のy座標の範囲 $y_{m1} \sim y_{m2}$ に入っているかどうかを判断する。範囲内でない場合は、ステップS 2 1 8に進み、範囲内の場合は、ステップS 2 1 6に進む。

5 ステップS 2 1 6にて、CPU 2 0 1は、敵オブジェクト1 1 5の座標を基に、エフェクト1 1 9の座標を算出する。なぜなら、 $x_{m1} < x_{pq} < x_{m2}$ 、を満たし、かつ、 $y_{m1} < y_{pq} < y_{m2}$ 、が満たされたということは、剣軌跡オブジェクト1 1 7が敵オブジェクト1 1 5にヒットしたということであるから、エフェクト1 1 9を発生させる必要があるからである。

ステップS 2 1 7にて、CPU 2 0 1は、スイング情報A 0 ~ A 7を基に、エフェクト1 1 9のアニメーションのためのアニメーションテーブル格納位置情報を登録する（ヒット登録：トリガに相当）。

10 ステップS 2 2 1にて、CPU 2 0 1は、スイング終了フラグをオフにする。

図4 7は、図3 8のステップS 6 5の盾検出処理の流れを示すフローチャートである。図4 7に示すように、ステップS 2 3 0にて、CPU 2 0 1は、注目点カウンタのカウンタ値 $c$ と、所定の閾値 $Th_A$ と、を比較する。

15 ステップS 2 3 1にて、CPU 2 0 1は、カウンタ値 $c$ が、所定の閾値 $Th_A$ より大きいと判断した場合、つまり、剣3の刀身部1 5の側面に取り付けられた反射シート1 7が検出された場合、ステップS 2 3 2に進む。

ステップS 2 3 2にて、CPU 2 0 1は、式 (1 4) 及び式 (1 5) により、盾オブジェクト1 2 3のx方向の移動距離 $l_x$ 及びy方向の移動距離 $l_y$ を求める。

20 ステップS 2 3 3にて、CPU 2 0 1は、式 (1 6) 及び式 (1 7) により、移動後の盾オブジェクト1 2 3の座標 ( $x_s, y_s$ ) を求める。

ステップS 2 3 4にて、CPU 2 0 1は、盾オブジェクト1 2 3のアニメーションのためのアニメーションテーブル格納位置情報を登録する（盾登録：トリガに相当）。

ステップS 2 3 5にて、CPU 2 0 1は、盾フラグをオンにする。

25 ステップS 2 4 2にて、CPU 2 0 1は、注目点カウンタ $c$ をリセットして、図3 8のステップS 6 6に進む。

さて、ステップS 2 3 1にて、CPU 2 0 1は、カウンタ値 $c$ が、所定の閾値 $Th_A$ 以下と判断した場合、つまり、剣3の刀身部1 5の側面に取り付けられた反射シート1 7が検出されなかった場合、ステップS 2 3 6に進む。

ステップS 2 3 6にて、CPU 2 0 1は、盾フラグがオンかどうかを判断する。盾フラグがオンならば、ステップS 2 3 7に進み、オフならば、ステップS 2 4 2に進む。

ステップS 2 3 7にて、CPU 2 0 1は、盾消滅カウンタeをインクリメントする。

5 ステップS 2 3 8にて、CPU 2 0 1は、盾消滅カウンタeが、所定値Eを下回っているかどうかを判断する。盾消滅カウンタeが、所定値Eを下回っている場合は、ステップS 2 4 2に進み、所定値E以上の場合は、ステップS 2 3 9に進む。つまり、ステップS 2 3 8では、盾フラグがオンになった後、剣3の側面の反射シート1 7が、E回連続して検出されなかった場合に、盾オブジェクト1 2 3を消滅させるべく、処理をステップS 2 3 9に進める。

10 ステップS 2 3 9にて、CPU 2 0 1は、盾オブジェクト1 2 3の表示座標をスクリーン9 1の範囲外とする（消滅登録）。これにより、スクリーン9 1に盾オブジェクト1 2 3が表示されない。

ステップS 2 4 0にて、CPU 2 0 1は、盾フラグをオフにする。ステップS 2 4 1にて、CPU 2 0 1は、盾消滅カウンタeをリセットする。

15 図4 8は、図3 8のステップS 6 6の説明進行処理の流れを示すフローチャートである。図4 8に示すように、ステップS 2 5 0にて、CPU 2 0 1は、画面に説明オブジェクト1 2 9が表示されているかどうかを判断する。説明オブジェクト1 2 9が表示されていない場合は、ステップS 2 5 4に進み、表示されている場合は、ステップS 2 5 1に進む。

ステップS 2 5 1にて、CPU 2 0 1は、角度フラグ及び方向フラグを参照して、剣3のスイングをチェックする。

20 CPU 2 0 1は、剣3が縦方向に振り下ろされた（スイング情報が「A 3」）場合は、ステップS 2 5 3に進み、それ以外では、ステップS 2 5 4に進む（ステップS 2 5 2）。

ステップS 2 5 3にて、CPU 2 0 1は、次の説明オブジェクト1 2 9を表示するためのアニメーションテーブル格納位置情報を登録する（説明進行登録：トリガに相当）。

ステップS 2 5 4にて、CPU 2 0 1は、角度フラグ及び方向フラグをリセットして、図3 8のステップS 6 7に処理を進める。

25 図4 9は、図3 8のステップS 6 7の前進処理の流れを示すフローチャートである。図4 9に示すように、ステップS 2 6 0にて、CPU 2 0 1は、スクリーン9 1に、前進を指示する説明オブジェクト1 3 2が表示されているかどうかを判断する。この説明オブジェクト1 3 2が表示されている場合は、ステップS 2 6 1に処理が進められ、表示されていない場合は、図3 8のステップS 6 8に処理が進められる。

ステップS 2 6 1にて、CPU 2 0 1は、所定フレーム数の間、剣3の注目点が画面の中心座標を中心にした所定範囲に存在するかどうかをチェックする。

CPU 2 0 1は、所定フレーム数の間、剣3の注目点が画面の中心座標を中心にした所定範囲に存在する場合は、ステップS 2 6 3へ進み、そうでない場合は、図3 8のステップS 6 8に処理が進められる（ステップS 2 6 2）。

ステップS 2 6 3にて、CPU 2 0 1は、仮想空間内を一定距離進むたびに、背景表示のための配列の全要素を更新する（前進登録）。

図5 0は、図3 8のステップS 7 0の画像情報セット処理の流れを示すフローチャートである。図5 0に示すように、ステップS 2 7 0では、CPU 2 0 1は、剣軌跡登録がなされている場合は、剣軌跡オブジェクト1 1 7に関する画像情報をセットする。具体的には、次の通りである。

CPU 2 0 1は、剣軌跡オブジェクト1 1 7の中心座標（ $x_t$ 、 $y_t$ ）と、剣軌跡オブジェクト1 1 7のサイズ情報と、スプライトのサイズ情報と、に基づいて、剣軌跡オブジェクト1 1 7を構成する各スプライトの座標を算出する。

また、CPU 2 0 1は、アニメーションテーブルを参照して、画像格納位置情報、コマ指定情報、及び、サイズ情報、を基に、表示すべき剣軌跡オブジェクト1 1 7の格納位置情報を算出する。さらに、CPU 2 0 1は、スプライトのサイズ情報を基に、表示すべき剣軌跡オブジェクト1 1 7を構成する各スプライトの格納位置情報を算出する。

ステップS 2 7 1では、CPU 2 0 1は、ヒット登録がなされている場合は、エフェクト1 1 9に関する画像情報をセットする。具体的には、次の通りである。

CPU 2 0 1は、エフェクト1 1 9の座標と、エフェクト1 1 9のサイズ情報と、スプライトのサイズ情報と、に基づいて、エフェクト1 1 9を構成する各スプライトの座標を算出する。

また、CPU 2 0 1は、アニメーションテーブルを参照して、画像格納位置情報、コマ指定情報、及び、サイズ情報、を基に、表示すべきエフェクト1 1 9の格納位置情報を算出する。さらに、CPU 2 0 1は、表示すべきエフェクト1 1 9を構成する各スプライトの格納位置情報を算出する。

ステップS 2 7 2では、CPU 2 0 1は、盾登録がなされている場合は、盾オブジェクト1 2 3に関する画像情報をセットする。具体的には、次の通りである。

CPU 2 0 1は、盾オブジェクト1 2 3の中心座標（ $x_s$ 、 $y_s$ ）と、盾オブジェクト1 2 3のサイズ情報と、スプライトのサイズ情報と、に基づいて、盾オブジェクト1 2 3を構成する各スプライトの座標を算出する。

また、CPU201は、アニメーションテーブルを参照して、画像格納位置情報、コマ指定情報、及び、サイズ情報、を基に、表示すべき盾オブジェクト123の格納位置情報を算出する。さらに、CPU201は、表示すべき盾オブジェクト123を構成する各スプライトの格納位置情報を算出する。

- 5 ステップS273では、CPU201は、スプライトからなる他のオブジェクト（例えば、説明オブジェクト129等）に関する画像情報（各スプライトの格納位置情報及び表示座標）をセットする。

図51は、図32のステップS5のモード選択処理の流れを示すフローチャートである。図51に示すように、ステップS300～ステップS302の処理は、それぞれ、図38のステップS60～ステップS62の処理と同様であり、説明を省略する。

ステップS303にて、CPU201は、カーソル101の移動処理を実行する。

- 10 図52は、図51のステップS303のカーソル移動処理の流れを示すフローチャートである。図52に示すように、ステップS320にて、CPU201は、剣3の注目点の座標を基に、カーソル101の座標を算出する。

ステップS321にて、CPU201は、カーソル101のアニメーションのためのアニメーションテーブル格納位置情報を登録する（カーソル登録）。

- 15 図51の説明に戻る。ステップS304にて、CPU201は、項目オブジェクト109の移動処理を実行する。

- 図53は、図51のステップS304の項目オブジェクト移動処理の流れを示すフローチャートである。図53に示すように、ステップS330にて、CPU201は、カーソル101が、図12の左移動指示オブジェクト103を中心としたレンジR1内に存在するかどうかを判断する。CPU201は、  
20 カーソル101がレンジR1内に存在する場合は、ステップS331に進み、存在しない場合は、ステップS332に進む。

ステップS331にて、CPU201は、項目オブジェクト109のx方向の速さ $v_x$ を「 $-v$ 」とする。

- 一方、ステップS332にて、CPU201は、カーソル101が、図12の右移動指示オブジェクト105を中心としたレンジR2内に存在するかどうかを判断する。CPU201は、カーソル101  
25 がレンジR2内に存在する場合は、ステップS334に進み、存在しない場合は、ステップS333に進む。

ステップS334にて、CPU201は、項目オブジェクト109のx方向の速さ $v_x$ を「 $v$ 」とする。



一方、ステップS 3 3 3にて、CPU 2 0 1は、項目オブジェクト1 0 9のx方向の速さ $v_x$ を「0」とする。

ステップS 3 3 5にて、CPU 2 0 1は、項目オブジェクト1 0 9のx座標に速さ $v_x$ を加えて、項目オブジェクト1 0 9の移動後のx座標とする。

- 5     ステップS 3 3 6にて、CPU 2 0 1は、項目オブジェクト1 0 9の表示のためのアニメーションテーブル格納位置情報を登録する（項目オブジェクト登録）。

図5 1の説明に戻る。ステップS 3 0 5及びステップS 3 0 6の処理は、それぞれ、図3 8のステップS 6 8及びステップS 6 9の処理と同様であり、説明を省略する。

- 10    ステップS 3 0 7にて、CPU 2 0 1は、カーソル1 0 1に関する画像情報をセットする。具体的には、次の通りである。

CPU 2 0 1は、カーソル1 0 1の座標と、カーソル1 0 1のサイズ情報と、スプライトのサイズ情報と、に基づいて、カーソル1 0 1を構成する各スプライトの座標を算出する。

- 15    また、CPU 2 0 1は、アニメーションテーブルを参照して、画像格納位置情報、コマ指定情報、及び、サイズ情報、を基に、表示すべきカーソル1 0 1の格納位置情報を算出する。さらに、CPU 2 0 1は、表示すべきカーソル1 0 1を構成する各スプライトの格納位置情報を算出する。

また、CPU 2 0 1は、項目オブジェクト1 0 9に関する画像情報をセットする。具体的には、次の通りである。

- 20    CPU 2 0 1は、項目オブジェクト1 0 9の座標と、項目オブジェクト1 0 9のサイズ情報と、スプライトのサイズ情報と、に基づいて、項目オブジェクト1 0 9を構成する各スプライトの座標を算出する。

また、CPU 2 0 1は、アニメーションテーブルを参照して、画像格納位置情報、コマ指定情報、及び、サイズ情報、を基に、表示すべき項目オブジェクト1 0 9の格納位置情報を算出する。さらに、CPU 2 0 1は、表示すべき項目オブジェクト1 0 9を構成する各スプライトの格納位置情報を算出する。

- 25    図5 4は、図3 2のステップS 6のスイング補正モードの流れを示すフローチャートである。図5 4に示すように、ステップS 4 0 0～ステップS 4 0 3の処理は、それぞれ、図3 8のステップS 6 0～ステップS 6 3の処理と同様であり、説明を省略する。

ステップS 4 0 4にて、CPU 2 0 1は、補正情報 $K_x$ 、 $K_y$ を取得する（図3 1参照）。

図5 5は、図5 4のステップS 4 0 4の補正情報取得処理の流れを示すフローチャートである。図5 5に示すように、ステップS 4 1 0にて、CPU 2 0 1は、角度フラグ及び方向フラグに基づいて、ス

イング情報を決定する（図23（a）～図23（c）参照）。そして、CPU201は、スイング情報が「A0」の場合は、ステップS411に進み、スイング情報が「A3」の場合は、ステップS412に進み、スイング情報がそれら以外の場合は、図54のステップS405に進む。

ステップS411では、剣3が横に振られているため、CPU201は、y方向の補正情報 $K_y$ を求める。

一方、ステップS412では、剣3が縦に振られているため、CPU201は、x方向の補正情報 $K_x$ を求める。

図54の説明に戻る。ステップS405、S406の処理は、それぞれ、図38のステップS68、S69の処理と同様であり、説明を省略する。

ステップS407にて、CPU201は、スイング補正画面（図31参照）を表示するための全スプライトの画像情報をセットする。

図56は、撮像ユニット5によるストロボスコープ撮影処理の流れを示すフローチャートである。ステップS500において、高速プロセッサ200は、ストロボスコープ撮影のために赤外発光ダイオード7を点灯する。具体的には、図10に示すLEDコントロール信号LEDCをHレベルとする。その後、ステップS501で、イメージセンサ43は、ピクセルデータを出力する。

ステップS502において、高速プロセッサ200は、ストロボスコープ撮影のために赤外発光ダイオード7を消灯する。具体的には、図10に示すLEDコントロール信号LEDCをLレベルとする。その後、ステップS503で、イメージセンサ43は、ピクセルデータを出力する。

以上の処理がゲームが終了するまで繰り返される（ステップS504）。

さて、次に、ゲーム画面の上記以外の例をいくつか挙げる。図57は、ゲーム画面の他の例示図である。図57に示すように、このゲーム画面には、人オブジェクト501及び動物オブジェクト502が表示される。そして、剣3の動きに応じて、カーソル503が移動する。このカーソル503を人オブジェクト501に重ねると、人オブジェクト501に関連付けられた説明オブジェクト500が表示される。一方、図示していないが、オペレータ94が、剣3を動かして、カーソル503を、動物オブジェクト502に重ねると、動物オブジェクト502に関連付けられた説明オブジェクトが表示される。

ここで、カーソル503の移動処理は、カーソル101の移動処理と同様である。そして、カーソル502が、人オブジェクト501を含む所定範囲内に移動してきた場合、人オブジェクト501に関連付けられた説明オブジェクト500が表示される。動物オブジェクト502についても同様である。

図58は、ゲーム画面のさらに他の例示図である。図58に示すように、このゲーム画面には、文字選択部505、選択枠506、左移動指示オブジェクト103、右移動指示オブジェクト105、文字表示部507、及び、カーソル101、が表示される。オペレータ94が剣3を操作してカーソル101を動かし、左移動指示オブジェクト103に重ねると、文字選択部505の文字が左方向に回転する。一方、右移動指示オブジェクト105に重ねると、文字選択部505の文字が右方向に回転する。このようにして、ア～ンまでの文字を選択できる。そして、剣3を、一定以上の速度で縦方向に振り下ろすと、選択枠506に入った文字が、文字表示部507に表示される。このようにして、オペレータ94は、剣3を操作して、文字表示部507に文字を表示することができる。

ここで、文字選択部505における文字の回転処理は、図12における項目オブジェクト109の回転処理と同様である。

図59は、ゲーム画面のさらに他の例示図である。図59に示すように、このゲーム画面には、斜め方向に炎オブジェクト510が表示されている。これは、オペレータ94が剣3を斜め方向に振ったことに応じて、表示されたものである。つまり、これまでの例では、オペレータ94が剣3を振ると、その動作に応じた剣軌跡オブジェクト117を表示していたが、その代わりに、炎オブジェクト510を表示している。炎オブジェクト510の表示のためのトリガ発生の処理は、剣軌跡オブジェクト117の表示のためのトリガ発生の処理と同様である。また、例えば、炎オブジェクト510は、注目点の座標に出現させる。

図60は、ゲーム画面のさらに他の例示図である。図60に示すように、このゲーム画面には、スイングガイド520、521、522および進行バー523が表示される。スイングガイド520～522においては、切り欠いた方向から剣3を振ることを意味する。オペレータ94は、進行バー523がスイングガイド520～522に重なったタイミングで、進行バー523が重なったスイングガイド520～522が指示する方向から、剣3を振る。図60の例では、オペレータ94は、進行バー523が重なったスイングガイド520が指示するように、左から横方向に、剣3を振る。

また、オペレータ94が、進行バー523が指示するタイミングで、かつ、スイングガイド520～522が指示する方向から、剣3を適正に振ることができれば、特別なオブジェクトを表示することもできる。

さて、図61(a)～図61(c)は、図1の剣3のさらに他の例示図である。図61(a)に示すように、この剣3の刀身部15の側面には、図2の反射シート17に代えて、円形の反射シート550及び反射シート551が、所定間隔で取り付けられている。従って、2点(反射シート550及び反射

シート 5 5 1) が検出された場合と、1 点 (半円柱状部材 2 1 に取り付けられた反射シート 2 3) が検出された場合と、でその後の処理を異ならせることができる。例えば、CPU 2 0 1 は、2 点検出の場合と、1 点検出の場合と、で異なる画像を、グラフィックプロセッサ 2 0 2 に表示させる。2 点検出については、後で詳細に説明する。なお、一方の半円柱状部材 2 1 に取り付けられた反射シート 2 3 と、  
5 他方の半円柱状部材 2 1 に取り付けられた反射シート 2 3 と、は近距離のため、イメージセンサ 4 3 には、1 点として撮影される。

また、図 6 1 (b) に示すように、この剣 3 の刀身部 1 5 の側面には、図 2 の反射シート 1 7 に代えて、長方形の反射シート 5 5 5 が取り付けられている。CPU 2 0 1 は、検出した反射シートの長辺と短辺との比を求めて、この比が所定値より大きければ、長方形の反射シート 5 5 5 が検出されたと判断  
10 する。従って、長方形の反射シート 5 5 5 が検出された場合と、反射シート 2 3 が検出された場合と、でその後の処理を異ならせることができる。例えば、CPU 2 0 1 は、検出した反射面に応じて、異なる画像を、グラフィックプロセッサ 2 0 2 に表示させる。

さらに、図 6 1 (c) に示すように、この剣 3 の刀身部 1 5 の側面には、図 2 の反射シート 1 7 に代えて、三角形の反射シート 5 6 0 が取り付けられている。CPU 2 0 1 は、検出した反射シートの形状  
15 を求めて、三角形であれば、反射シート 5 6 0 が検出されたと判断する。従って、三角形の反射シート 5 6 0 が検出された場合と、反射シート 2 3 が検出された場合と、でその後の処理を異ならせることができる。例えば、CPU 2 0 1 は、検出した反射面に応じて、異なる画像を、グラフィックプロセッサ 2 0 2 に表示させる。

なお、図 6 1 (a) ~ 図 6 1 (c) において、半円柱状部材 2 1 及び反射シート 2 3 を剣 3 に設ける  
20 代わりに、図 4 及び図 5 の反射シート 3 1 を、剣 3 の刃先部に設けることもできる。

上記では、剣型の操作物 3 を例に挙げた。次に、剣型以外の操作物 3 の一例を説明する。図 6 2 は、オペレータ 9 4 によって操作される操作物の例示図である。この操作物 3 は、スティック 5 7 0 の両端に球状部材 5 7 1、5 7 2 を取り付けたものである。球状部材 5 7 1、5 7 2 には、反射シート 5 7 5、5 7 6 が取り付けられる。オペレータ 9 4 は、スティック 5 7 0 を握って、操作物 3 を操作する。2 つ  
25 の反射シート 5 7 5、5 7 6 は、所定間隔で取り付けられているため、イメージセンサ 4 3 により、2 つの注目点が撮影される。CPU 2 0 1 は、2 つの反射シート 5 7 5、5 7 6 の状態情報を求める。そして、CPU 2 0 1 は、その 2 つの反射シート 5 7 5、5 7 6 の状態情報に応じて、グラフィックプロセッサ 2 0 2 に画像を表示させる。

次に、図6 1 (a) 及び図6 2 で実行する2点の抽出処理について説明する。この場合、一方の反射シートを第1の反射シートと呼び、他方の反射シートを第2の反射シートと呼ぶ。

図6 3は、第1の反射シートの注目点（第1注目点）の座標算出の説明図である。図6 3に示すように、イメージセンサ4 3は、例えば、3 2ピクセル×3 2ピクセルからなる。CPU 2 0 1は、Y方向に3 2ピクセル分の差分データをスキャンし、そして、X座標をインクリメントし、Y方向に3 2ピクセル分の差分データをスキャンし、そして、X座標をインクリメントし、というように、X座標をインクリメントしながら、Y方向（列方向）に3 2ピクセル分の差分データをスキャンしていく。

この場合、CPU 2 0 1は、Y方向にスキャンした3 2ピクセル分の差分データから、最大輝度値の差分データを求め、その最大輝度値と所定の閾値 $T_h$ とを比較する。そして、CPU 2 0 1は、その最大輝度値が所定の閾値 $T_h$ より大きい場合は、その値を配列 $max[n]$ に代入する。一方、CPU 2 0 1は、その最大輝度値が所定の閾値 $T_h$ 以下の場合は、所定値（例えば、「0」）を配列 $max[n]$ に代入する。

ここで、 $n$ はX座標であり、さらに、CPU 2 0 1が、最大輝度値を有するピクセルのY座標を関連付けて格納することで、後に、最大輝度値を有するピクセルのX座標及びY座標を取得できる。

さらに、CPU 2 0 1は、配列 $max[0] \sim max[31]$ をスキャンして、さらにその中の最大値を求める。そして、CPU 2 0 1は、その最大値のX座標及びY座標を、第1の反射シートの注目点の座標（ $X_1, Y_1$ ）として保存する。

次に、第2の反射シートの注目点（第2注目点）の座標算出について説明する。CPU 2 0 1は、配列 $max[0] \sim max[31]$ のうちの最大値、つまり、第1の反射シートの注目点の座標（ $X_1, Y_1$ ）に位置するピクセルの差分データ、を中心に所定範囲をマスクする。この点を図面を用いて説明する。

図6 4は、第2の反射シートの注目点の座標算出の説明図である。図6 4に示すように、CPU 2 0 1は、配列 $max[0] \sim max[31]$ のうちの最大値（図6 4の例では、 $X=9, Y=9$ ）を中心に所定範囲をマスク（太枠で囲った部分）する。

そして、CPU 2 0 1は、このマスクした範囲を除いて、配列 $max[0] \sim max[31]$ をスキャンする。つまり、この例では、CPU 2 0 1は、配列 $max[0] \sim max[6]$ 、及び、配列 $max[12] \sim max[31]$ をスキャンする。

そして、CPU 2 0 1は、そのスキャンした配列 $max[0] \sim max[6]$ 、及び、配列 $max[12] \sim max[31]$ のうちから、最大値を求める。CPU 2 0 1は、求めた最大値のX座標及

びY座標を、第2の反射シートの注目点の座標(X2, Y2)として保存する。図64の例では、最大値は、配列max[22]であり、従って、第2の反射シートの注目点の座標は、X2=22、Y2=10、である。なお、図64の例では、第1の反射シートの注目点の座標は、X1=9、Y1=9、となっている。

- 5      なお、第1注目点及び第2注目点の座標を求める際の最大値の検出は、実際は、スキャンしながら行われる。上記では、説明の便宜のため、スキャンした後に、最大値を求めるように記載している。

さて、以上のように本実施の形態によれば、ストロボスコープにより間欠的に光が照射される剣3をイメージセンサ43により撮影して、剣3の状態情報を求めている。このように、実空間に検知面(二次元)を形成することなく、イメージセンサ43の撮影範囲である検知空間(三次元)に存在する剣3  
10    の状態情報を取得できる。従って、剣3の操作範囲が二次元平面に制限されることがないため、オペレータ94による剣3の操作の制約が小さく、操作の自由度を大きくできる。

また、テレビジョンモニタ90のスクリーン91に対応した検知面を実空間に形成する必要もないため、設置場所の制限を小さくできる(省スペースの実現)。

- さらに、剣3を振ったことに基づくトリガ(剣軌跡登録に相当)により、剣3の移動軌跡を表現する  
15    剣軌跡オブジェクト117がスクリーン91に表示される。このため、オペレータ94は、現実に見ることができない移動軌跡を、スクリーン91上で見ることができ、剣3を振った実感をより味合うことができる。

- この場合、フレームごとに幅の異なる帯状オブジェクトを表示することにより、剣3の移動軌跡を表現する。このときの帯状オブジェクトの幅は、フレームの更新とともに太くなった後、フレームの更新  
20    とともに細くなる(図27～図29参照)。

このため、鋭い閃光が走ったような、剣3の移動軌跡を表示できる。特に、帯状オブジェクトの色を工夫することで、その効果をより高めることができる。

- さらに、スクリーン91に表示される仮想世界に、オペレータ94が操作した剣3の移動軌跡が出現するため、このような剣3の移動軌跡の表示を通じて、オペレータ94は、仮想世界との接触が可能となり、仮想世界をより楽しむことができる。つまり、オペレータ94は、あたかも、スクリーン91に  
25    表示されるゲーム世界で、ゲームを楽しんでいるような実感を得ることができる。

さらに、撮像ユニット5により検知された反射面(例えば、反射シート17, 23)に応じて、異なる画像(例えば、剣軌跡オブジェクト117、盾オブジェクト123)が表示されるため、単一の操作物3を操作するだけで、反射面の数に応じた、異なる画像を表示できる。このため、異なる画像ごとに、



22～図26参照)。このため、剣3の移動軌跡をよりの確に反映した剣軌跡オブジェクト117の態様を決定できる。

なお、剣3の近接する2つの注目点に基づいて、剣軌跡オブジェクト117の態様を決定すると、例えば、次の不都合がある。オペレータ94が、自分の感覚で直線的に剣3を移動させた場合でも、実際には、若干の円弧を描いている場合もある。この場合には、当然、イメージセンサ43には、円弧を描くように剣3が撮影される。このときに、近接する2つの注目点に基づいて、剣軌跡オブジェクト117の態様を決定すると、オペレータ94の感覚とずれた態様の剣軌跡オブジェクト117が表示されることになる。例えば、剣3を横に振ったつもりが、斜め方向の剣軌跡オブジェクト117が表示されるような事態である。

- 10 さらに、剣3の状態情報に基づくトリガ（説明進行登録に相当）により、スクリーン91に文字列を次々に表示させることができるため、文字列の更新に使用するスイッチやアナログスティック等を剣3に設ける必要もなく、剣3の製造コストを低減できるばかりでなく、操作性をも向上できる（図17参照）。

- 15 さらに、剣3の状態情報に基づくトリガ（進行登録に相当）により、背景を更新することができるため、背景の更新に使用するスイッチやアナログスティック等を剣3に設ける必要もなく、剣3の製造コストを低減できるばかりでなく、操作性をも向上できる（図18参照）。

- さらに、CPU201は、剣3の位置情報を補正する補正情報 $K_x$ 、 $K_y$ を取得する。そして、CPU201は、補正情報 $K_x$ 、 $K_y$ を用いて、補正後の位置情報を算出する。このため、オペレータ94が剣3を操作する感覚と、CPU201が算出する剣3の位置情報と、のずれを極力解消できるため、  
20 オペレータ94による剣3の操作をよりの確に反映した画像を表示できる。

さらに、剣3の位置情報に基づいて、カーソル101を移動させることができるため、カーソル101の移動に使用するスイッチやアナログスティック等を剣3に設ける必要もなく、剣3の製造コストを低減できるばかりでなく、操作性をも向上できる（図12参照）。

- さらに、剣3の状態情報に基づいて、予め定められた処理を実行することを確定する。例えば、剣3  
25 を一定以上の速さで、縦方向に振り下ろすことで、項目オブジェクト109の選択が確定され、選択された項目に応じた処理の実行が開始される（図12参照）。このように、剣3の状態情報に基づいて、処理の実行の確定ができるため、処理の実行の確定に使用するスイッチやアナログスティック等を剣3に設ける必要もなく、剣3の製造コストを低減できるばかりでなく、操作性をも向上できる。



さらに、カーソル503が人オブジェクト501に重なったときに、その人オブジェクト501に関連付けられた説明オブジェクト500が表示される（図57参照）。このため、オペレータ94は、剣3の操作によりカーソル503を移動させるだけで、表示されている人オブジェクト501に関連付けられた画像を表示させることができる。

- 5 さらに、カーソル101により選択された文字をスクリーン91に表示できる（図58参照）。このため、オペレータ94は、剣3の操作によりカーソル101を移動させて所望の文字を選択するだけで、文字を入力できるため、文字入力に使用するスイッチやアナログスティック等を剣3に設ける必要もなく、剣3の製造コストを低減できるばかりでなく、操作性をも向上できる。

- 10 さらに、剣3の状態情報に基づくトリガにより、剣3の動きに応じた炎オブジェクト510をスクリーン91に表示できる。これにより、剣3の移動軌跡を表現する剣オブジェクト117と異なった視覚的効果を、オペレータ94に与えることができる（図59参照）。

- さらに、剣軌跡登録が行われてから（トリガが発生してから）、所定時間経過後に（人間の感覚で）、剣3の移動軌跡を表現する剣軌跡オブジェクト117をスクリーン91に表示することもできる。この場合、剣軌跡オブジェクト117を、剣軌跡登録（トリガの発生）とほぼ同時（人間の感覚で同時）に  
15 表示する場合と比較して、異なった効果を、オペレータ94に与えることができる。

- さらに、剣3の連続した状態情報が所定条件（例えば、剣3を、縦→横→縦、に連続して振る、等）を満足したときに、所定のオブジェクトを表示することもできる。これにより、剣3の操作が所定条件を満足したときにのみ、所定のオブジェクトが表示されるため、この所定条件の設定の仕方によって、所定のオブジェクトを表示させるための、オペレータ94による剣3の操作を任意にコントロールでき  
20 る。

さらに、剣3の操作方向を指示するガイドオブジェクト520～522及び操作タイミングを指示する進行バー523を表示することもできる。この場合は、オペレータは、情報処理装置1が要求する剣3の操作方向及び操作タイミングを視覚的に認識できる。

- さらに、CPU201は、状態情報として、速さ情報、移動方向情報、移動距離情報、速度ベクトル  
25 情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、それらのうちのいくつか、又は、それらの全部、を算出できる。このため、オペレータ94による剣3の様々な動きに応じたオブジェクトをスクリーン91に表示できる。

さらに、剣軌跡登録（トリガ）に基づいて、テレビジョンモニタ90のスピーカから効果音を発生させることができる。このため、オペレータ94に対して、視覚的効果に加えて、聴覚的効果を与えるこ

とができる。従って、オペレータ 9 4 は、スクリーン 9 1 に表示される仮想世界をより一層楽しむことができる。例えば、オペレータ 9 4 が操作した剣 3 の移動軌跡 1 1 7 が仮想世界に出現すると同時に、効果音を発生させれば、オペレータ 9 4 は、仮想世界をより一層楽しむことができる。

さらに、操作物 3 の複数の反射シート 5 7 5, 5 7 6 の状態情報に応じて画像を表示することもできるため、単一の反射シート 5 7 5 の状態情報に応じて画像を表示する場合と比較して、操作物 3 の状態をより反映した画像を表示できる（図 6 2 参照）。

さらに、発光時画像信号と消灯時画像信号との差分信号を生成するといった簡単な処理のみで、ノイズや外乱の影響を抑えた精度の高い検出が可能であるので、コスト、許容される消費電力等の条件により情報処理装置 1 のパフォーマンスが制限されるシステムの上でも容易に達成が可能である。

10      なお、本発明は、上記の実施の形態に限られるものではなく、その要旨を逸脱しない範囲で種々の態様において実施することが可能であり、例えば、以下のような変形も可能である。

（1）実施の形態では、剣型の操作物 3 を例に挙げたが（図 2、図 4、図 6 1）、本発明は、これに限定されない。また、本発明は、図 6 2 の操作物 3 にも限定されない。つまり、光を反射できる部材（例えば、再帰反射シート）を備えていれば、任意の形状の操作物 3 を使用できる。

15      （2）実施の形態では、図 2 7～図 2 9 のアニメーションにより、剣軌跡オブジェクト 1 1 7 を表現したが、本発明は、これに限定されない。

（3）実施の形態では、2 種類の反射面（例えば、図 2 の反射シート 1 7, 2 3）を操作物 3 に設けたが、1 種類でもよく、また、3 種類以上でもよい。

20      （4）図 7 の高速プロセッサ 2 0 0 として、任意の種類のプロセッサを使用できるが、本件出願人が既に特許出願している高速プロセッサ（商品名：X a v i X）を用いることが好ましい。この高速プロセッサは、例えば、特開平 1 0－3 0 7 7 9 0 号公報およびこれに対応するアメリカ特許第 6, 0 7 0, 2 0 5 号に詳細に開示されている。

以上、本発明を実施例により詳細に説明したが、当業者にとっては、本発明が本願中に説明した実施例に限定されるものではないということは明らかである。本発明は、特許請求の範囲の記載により定まる本発明の趣旨及び範囲を逸脱することなく修正及び変更態様として実施することができる。従って、  
25      本願の記載は、例示説明を目的とするものであり、本発明に対して何ら制限的な意味を有するものではない。

## 請求の範囲

1. オペレータに保持されて動きが与えられる操作物の動きを加味した画像をディスプレイ装置に表示させる情報処理装置であって、
- 5 反射面を有する前記操作物に、予め定められた周期で、光を照射するストロボスコープと、  
前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、  
前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、  
前記差分信号を基に、前記操作物の状態情報を算出し、その状態情報に基づいて、第1のトリガを発生する状態情報算出ユニットと、  
10 前記第1のトリガに基づいて、前記操作物の移動軌跡を表現する第1のオブジェクトを前記ディスプレイ装置に表示させる画像表示処理ユニットと、を備える情報処理装置。
2. オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置に画像を表示させる情報処理装置であって、
- 15 複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するストロボスコープと、  
前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、  
前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、  
前記差分信号を基に、前記操作物の状態情報を算出して、その状態情報に基づいて、前記複数の反射  
20 面のいずれの反射面が撮影されたかを判別する状態情報算出ユニットと、  
判別された前記反射面に応じて、前記ディスプレイ装置に異なる画像を表示させる画像表示処理ユニットと、を備える情報処理装置。
3. 前記状態情報は、前記反射面の面積情報、数情報、形状情報、若しくは、形状を表す比率情報、のいずれか、又は、それらの組み合わせ、である、請求項2記載の情報処理装置。
- 25 4. オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置に画像を表示させる情報処理装置であって、  
複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するストロボスコープと、  
前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、

前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、  
前記差分信号に基づいて、前記反射面ごとの状態情報を算出する状態情報算出ユニットと、  
前記複数の反射面の前記状態情報に応じて、画像を表示させる画像表示処理ユニットと、を備える情報処理装置。

- 5 5. 前記移動軌跡を表現する前記第1のオブジェクトは、帯状オブジェクトであり、  
前記画像表示処理ユニットは、フレームごとに幅の異なる前記帯状オブジェクトを前記ディスプレイ装置に表示することにより、前記操作物の移動軌跡を表現し、  
前記帯状オブジェクトの前記幅は、フレームの更新とともに太くなった後、フレームの更新とともに細くなる、請求項1記載の情報処理装置。
- 10 6. 前記画像表示処理ユニットは、第2のオブジェクトを前記ディスプレイ装置に表示し、  
前記状態情報算出ユニットは、前記第2のオブジェクトと、前記操作物の移動軌跡を表現する前記第1のオブジェクトと、の位置関係が所定条件を満たしたときに、第2のトリガを発生し、  
前記画像表示処理ユニットは、前記第2のトリガに基づいて、予め定められたエフェクトを与えた前記第2のオブジェクトを前記ディスプレイ装置に表示する、請求項5記載の情報処理装置。
- 15 7. 前記状態情報算出ユニットは、  
前記操作物の前記状態情報としての速さ情報が、予め定められた第1の閾値を超えてから、予め定められた第2の閾値を下回るまでの、前記操作物の前記状態情報としての位置情報を算出し、あるいは、  
前記操作物の前記速さ情報が、前記予め定められた第1の閾値を超えてから、前記撮像ユニットによる撮影範囲を超える前までの、前記操作物の前記位置情報を算出し、かつ、
- 20 前記操作物の前記位置情報の取得回数が3以上の場合に、前記操作物の最初の前記位置情報と、前記操作物の最後の前記位置情報と、に基づいて、前記操作物の移動軌跡を表現する前記第1のオブジェクトの態様を決定し、かつ、  
前記操作物の前記位置情報の取得回数が3以上の場合に、前記状態情報に基づいて、前記第1のトリガを発生する、請求項1記載の情報処理装置。
- 25 8. 前記状態情報算出ユニットは、前記操作物の前記状態情報としての面積情報を算出し、その面積情報が、予め定められた第3の閾値を超えたときに、第3のトリガを発生し、  
前記画像表示処理ユニットは、前記第3のトリガに基づいて、第3のオブジェクトを前記ディスプレイ装置に表示する、請求項1記載の情報処理装置。
9. 前記画像表示処理ユニットは、文字列を前記ディスプレイ装置に表示し、

前記状態情報算出ユニットは、前記操作物の前記状態情報に基づいて、第4のトリガを発生し、

前記画像表示処理ユニットは、前記第4のトリガに基づいて、前記文字列と異なる文字列を前記ディスプレイ装置に表示する、請求項1記載の情報処理装置。

10. 前記状態情報算出ユニットは、前記操作物の前記状態情報に基づいて、第5のトリガを発生し、

5 前記画像表示処理ユニットは、前記第5のトリガに基づいて、背景画像を更新する、請求項1記載の情報処理装置。

11. 前記操作物の前記状態情報としての位置情報を補正する補正情報を取得する補正情報取得ユニット、をさらに備え、

10 前記状態情報算出ユニットは、前記補正情報を用いて、補正後の位置情報を算出する、請求項1記載の情報処理装置。

12. 前記画像表示処理ユニットは、カーソルを前記ディスプレイ装置に表示し、かつ、前記操作物の前記状態情報としての位置情報に応じて、前記カーソルを移動させる、請求項1記載の情報処理装置。

13. 前記操作物の前記状態情報に基づいて、予め定められた処理を実行することを確定する、請求項1記載の情報処理装置。

15 14. 前記画像表示処理ユニットは、前記カーソルが第4のオブジェクトに重ねて表示されたときに、その第4のオブジェクトに関連付けられた画像を前記ディスプレイ装置に表示する、請求項12記載の情報処理装置。

15. 前記画像表示処理ユニットは、前記カーソルにより選択された文字を前記ディスプレイ装置に表示する、請求項12記載の情報処理装置。

20 16. 前記状態情報算出ユニットは、前記操作物の前記状態情報に基づいて、第6のトリガを発生し、前記画像表示処理ユニットは、前記第6のトリガに基づいて、前記操作物の動きに応じた第5のオブジェクトを前記ディスプレイ装置に表示する、請求項1記載の情報処理装置。

25 17. 前記画像表示処理ユニットは、前記第1のトリガが発生してから、所定時間経過後に、前記操作物の移動軌跡を表現する前記第1のオブジェクトを前記ディスプレイ装置に表示する、請求項1記載の情報処理装置。

18. 前記画像表示処理ユニットは、前記操作物の連続した前記状態情報が所定条件を満足したときに、第6のオブジェクトを表示する、請求項1記載の情報処理装置。

19. 前記画像表示処理ユニットは、前記操作物の操作方向及び操作タイミングを指示するガイドを表示する、請求項1記載の情報処理装置。

20. 前記操作物の前記状態情報は、速さ情報、移動方向情報、移動距離情報、速度ベクトル情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、又は、それらの2以上の組み合わせ、である、請求項1記載の情報処理装置。
21. 前記第1のトリガに基づいて、スピーカから効果音を発生させる効果音生成ユニット、をさらに備える、請求項1記載の情報処理装置。
22. オペレータに保持されて動きが与えられ、反射面を有する操作物と、  
前記操作物に、予め定められた周期で、光を照射するストロボスコープと、  
前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、
- 10 前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、  
前記差分信号を基に、前記操作物の状態情報を算出し、その状態情報に基づいて、第1のトリガを発生する状態情報算出ユニットと、  
前記第1のトリガに基づいて、前記操作物の移動軌跡を表現する第1のオブジェクトを前記ディスプレイ装置に表示させる画像表示処理ユニットと、を備える、情報処理システム。
- 15 23. オペレータに保持されて動きが与えられ、複数の反射面を有する操作物と、  
前記操作物に、予め定められた周期で、光を照射するストロボスコープと、  
前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、  
前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、
- 20 前記差分信号を基に、前記操作物の状態情報を算出して、その状態情報に基づいて、前記複数の反射面のいずれの反射面が撮影されたかを判別する状態情報算出ユニットと、  
判別された前記反射面に応じて、前記ディスプレイ装置に異なる画像を表示させる画像表示処理ユニットと、を備える情報処理システム。
24. オペレータに保持されて動きが与えられ、複数の反射面を有する操作物と、
- 25 前記操作物に、予め定められた周期で、光を照射するストロボスコープと、  
前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得する撮像ユニットと、  
前記発光時画像と前記消灯時画像との差分信号を生成する差分信号生成ユニットと、  
前記差分信号に基づいて、前記反射面ごとの状態情報を算出する状態情報算出ユニットと、

前記複数の反射面の前記状態情報に応じて、画像を表示させる画像表示処理ユニットと、を備える情報処理システム。

25. 請求項2記載の情報処理装置のオペレータが操作する操作物であって、異なる複数の反射面を有する操作物。

5 26. オペレータに保持されて動きが与えられる操作物の動きを加味した画像をディスプレイ装置に表示させる情報処理方法であって、

反射面を有する前記操作物に、予め定められた周期で、光を照射するステップと、

前記光の発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得するステップと、

10 前記発光時画像と前記消灯時画像との差分信号を生成するステップと、

前記差分信号を基に、前記操作物の状態情報を算出し、その状態情報に基づいて、第1のトリガを発生するステップと、

前記第1のトリガに基づいて、前記操作物の移動軌跡を表現する第1のオブジェクトを前記ディスプレイ装置に表示させるステップと、を含む情報処理方法。

15 27. オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置に画像を表示させる情報処理方法であって、

複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するステップと、

前記光の発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得するステップと、

20 前記発光時画像と前記消灯時画像との差分信号を生成するステップと、

前記差分信号を基に、前記操作物の状態情報を算出して、その状態情報に基づいて、前記複数の反射面のいずれの反射面が撮影されたかを判別するステップと、

判別された前記反射面に応じて、前記ディスプレイ装置に異なる画像を表示させるステップと、を含む情報処理方法。

25 28. オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置に画像を表示させる情報処理方法であって、

複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するステップと、

前記光の発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像を取得するステップと、

前記発光時画像と前記消灯時画像との差分信号を生成するステップと、

前記差分信号に基づいて、前記反射面ごとの状態情報を算出するステップと、

前記複数の反射面の前記状態情報に応じて、画像を表示させるステップと、を含む情報処理方法。

29. オペレータに保持されて動きが与えられる操作物の動きを加味した画像をディスプレイ装置に表

5 示させるコンピュータに用いられる情報処理プログラムであって、

反射面を有する前記操作物に、予め定められた周期で、光を照射するステップと、

前記光の発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像  
を取得するステップと、

前記発光時画像と前記消灯時画像との差分信号を生成するステップと、

10 前記差分信号を基に、前記操作物の状態情報を算出し、その状態情報に基づいて、第1のトリガを発生するステップと、

前記第1のトリガに基づいて、前記操作物の移動軌跡を表現する第1のオブジェクトを前記ディスプレイ装置に表示させるステップと、を前記コンピュータに実行させる情報処理プログラム。

30. オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置

15 に画像を表示させるコンピュータに用いられる情報処理プログラムであって、

複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するステップと、

前記光の発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像  
を取得するステップと、

前記発光時画像と前記消灯時画像との差分信号を生成するステップと、

20 前記差分信号を基に、前記操作物の状態情報を算出して、その状態情報に基づいて、前記複数の反射面のいずれの反射面が撮影されたかを判別するステップと、

判別された前記反射面に応じて、前記ディスプレイ装置に異なる画像を表示させるステップと、を前記コンピュータに実行させる情報処理プログラム。

31. オペレータに保持されて動きが与えられる操作物を検出した結果に基づいて、ディスプレイ装置

25 に画像を表示させるコンピュータに用いられる情報処理プログラムであって、

複数の反射面を有する前記操作物に、予め定められた周期で、光を照射するステップと、

前記光の発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像及び消灯時画像  
を取得するステップと、

前記発光時画像と前記消灯時画像との差分信号を生成するステップと、



前記差分信号に基づいて、前記反射面ごとの状態情報を算出するステップと、

前記複数の反射面の前記状態情報に応じて、画像を表示させるステップと、を前記コンピュータに実行させる情報処理プログラム。

32. ゲームを行うためのゲームシステムであって、

5 オペレータによって実際に動かされる操作物と、

オペレータによって動かされている操作物を撮影するイメージセンサと、

ゲームを行う際にディスプレイ装置に接続され、前記イメージセンサからイメージ信号を受け取り、前記ゲームのコンテンツを前記ディスプレイ装置に表示する処理装置と、を備え、

前記操作物は、前記イメージセンサによって撮影された前記操作物のイメージに基づいて、前記ゲー

10 ムにおいて所定の役割を担っており、

前記ゲームを行っている際に、前記処理装置によって前記ディスプレイ装置に表示される前記ゲームのコンテンツにおける、前記操作物の移動軌跡の表示は、帯状イメージに簡略化されており、

この帯状イメージは、前記オペレータによって動かされる前記操作物の、前記ディスプレイ装置上の表示における移動軌跡の少なくとも2点を結ぶものであり、この少なくとも2点は、前記イメージセン

15 サで撮影されたイメージから取得される、ゲームシステム。

## 要約書

- 赤外発光ダイオード（７：図２）により、間欠的に赤外光が照射される剣（３：図２）を撮像ユニット（５：図２）により撮影して、剣の動きを検出する。剣が振られたことを検知したことをトリガとして、剣の移動軌跡を表す剣軌跡オブジェクト（１１７：図１４）をテレビジョンモニタ（９０：図１）に表示する。
- 5